DOCTOR OF PHILOSOPHY

Image analysis and representation for textile design classification

Jia, Wei

Award date:
2011

DOCTOR OF PHILOSOPHY

Image analysis and representation for textile design classification

Wei Jia

2011

University of Dundee

Image analysis and representation for textile design classification

A thesis submitted in application for the degree of Doctor of Philosophy

Wei Jia

School of Computing

University of Dundee

April 2011

Contents

List of Figures

List of Tables

Acknowledgements

I would like to acknowledge several people whose contributions greatly helped me to accomplish the work reported in this thesis. My supervisor, Professor Stephen J. McKenna, also the principal investigator of FABRIC project, provided me with essential guidance throughout countless regular meetings, in which I could benefit from his deep theoretical knowledge. His true scientific intuition has made him a sustainable source of idea which inspired and enrich my development as a PhD student and a researcher. His thorough and rigorous comments led to major improvements in all the manuscripts (including that of this thesis), presentations, and posters that I had to prepare. Anna Buruma, Martin Coward, and Peter Taylor, from Liberty Art Fabric provided experimental data and helpful discussions and advice. My Project Manager, Dr. Annette Ward, annotated the experimental metadata, helped to clarify and analyse the properties of the textile designs, and provided me valuable comments on writing, presentation, and posters. My second supervisor, Professor Ian W. Ricketts, contributed with many useful discussions and taught me the skills about how to do the academic presentation. Dr. Keith J. Edwards, clarified some basic graph theories and contributed many interesting ideas and indispensable discussions. My colleagues at the Computer Vision and Imaging Processing group (CVIP) have always ensured a friendly and stimulating work environment. Their choices of papers for our fortnightly reading group meetings often provided me with very useful ideas for this work. My family and friends offered essential support and encouragement throughout my studies,

and demonstrated great understanding for my limited availability to spend time with them. I should also acknowledge some organisations, Liberty Art Fabric, System Simulation, Calico Jack Ltd., and the Victoria and Albert Museum, in particular UK Technology Strategy Board[1] which offered the research grant and British Machine Vision Association (BMVA) which contributed with travel funds that allowed me to participate in several events of great relevance to this work. To all, my big Thanks.

[1]The Technology Strategy Board is a business-led executive non-departmental public body, established by the government. Its mission is to promote and support research into, and development and exploitation of, technology and innovation for the benefit of UK business, in order to increase economic growth and improve the quality of life. It is sponsored by the Department for Innovation, Universities and Skills (DIUS). Please visit `http://www.innovateuk.org/` for further information.

Associated publications

This work has been reported in research papers submitted to a number of events, namely: the International Conference on Computer Vision Theory and Applications (VISAPP) in 2009; the International Conference on Pattern Recognition (ICPR) in 2010; and the British Machine Vision Conference (BMVC) in 2010. These publications are listed below.

Additional dissemination material included a poster presented at the BMVA Computer Vision Summer School in 2008, and a presentation at the Scottish Informatics & Computer Science Alliance (SICSA) PhD Conference in 2009.

W. Jia, S. J. McKenna, and A. A. Ward. Extracting printed designs and woven patterns from textile images. In *Proceedings of International Conference on Computer Vision Theory and Application (VISAPP)*, *vol 1*, pp. 201-208, Lisboa, Portugal, 2009.

W. Jia, and S. J. McKenna. Classifying textile designs using bags of shapes, In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pp. 294-297, Istanbul, Turkey, 2010.

W. Jia, S. J. McKenna, A. A. Ward, and K. Edwards. Classifying textile designs using region graphs, In Proceedings of *the British Machine Vision Conference (BMVC)*, pp. 93.1-93.10, Aberystwyth, Wales, 2010.

Declaration by the author

I hereby declare that I am the author of this thesis; that all references cited have been consulted by me; that the work of which this thesis is a record has been done by me, and that it has not been previously accepted for a higher degree.

Signed

Wei Jia

April 2011

Declaration by the supervisor

I hereby certify that Mrs Wei Jia has satisfied all the terms and conditions of the regulations made under Ordinances 12 and 39 and has completed the required nine terms of research to qualify in submitting this thesis in application for the degree of Doctor of Philosophy.

Signed

Prof. Stephen McKenna

April 2011

Abstract

A good image representation is vital for image comparision and classification; it may affect the classification accuracy and efficiency. The purpose of this thesis was to explore novel and appropriate image representations. Another aim was to investigate these representations for image classification. Finally, novel features were examined for improving image classification accuracy. Images of interest to this thesis were textile design images. The motivation of analysing textile design images is to help designers browse images, fuel their creativity, and improve their design efficiency. In recent years, bag-of-words model has been shown to be a good base for image representation, and there have been many attempts to go beyond this representation. Bag-of-words models have been used frequently in the classification of image data, due to good performance and simplicity. "Words" in images can have different definitions and are obtained through steps of feature detection, feature description, and codeword calculation. The model represents an image as an orderless collection of local features. However, discarding the spatial relationships of local features limits the power of this model. This thesis exploited novel image representations, bag of shapes and region label graphs models, which were based on bag-of-words model. In both models, an image was represented by a collection of segmented regions, and each region was described by shape descriptors. In the latter model, graphs were constructed to capture the spatial information between groups of segmented regions and graph features were calculated based on some graph theory. Novel elements include use of MRFs

to extract printed designs and woven patterns from textile images, utilisation of the extractions to form bag of shapes models, and construction of region label graphs to capture the spatial information.

The extraction of textile designs was formulated as a pixel labelling problem. Algorithms for MRF optimisation and re-estimation were described and evaluated. A method for quantitative evaluation was presented and used to compare the performance of MRFs optimised using $\alpha-$expansion and iterated conditional modes (ICM), both with and without parameter re-estimation. The results were used in the formation of the bag of shapes and region label graphs models. Bag of shapes model was a collection of MRFs' segmented regions, and the shape of each region was described with generic Fourier descriptors. Each image was represented as a bag of shapes. A simple yet competitive classification scheme based on nearest neighbour class-based matching was used. Classification performance was compared to that obtained when using bags of SIFT features.

To capture the spatial information, region label graphs were constructed to obtain graph features. Regions with the same label were treated as a group and each group was associated uniquely with a vertex in an undirected, weighted graph. Each region group was represented as a bag of shape descriptors. Edges in the graph denoted either the extent to which the groups' regions were spatially adjacent or the dissimilarity of their respective bags of shapes. Series of unweighted graphs were obtained by removing edges in order of weight. Finally, an image was represented using its shape descriptors along with features derived from the chromatic numbers or domination numbers of the unweighted graphs and their complements. Linear SVM classifiers were used for classification.

Experiments were implemented on data from Liberty Art Fabrics, which consisted of more than 10,000 complicated images mainly of printed textile designs and woven patterns. Experimental data was classified into seven classes manually by assigning each image a text descriptor based on content or design type. The

seven classes were floral, paisley, stripe, leaf, geometric, spot, and check. The result showed that reasonable and interesting regions were obtained from MRF segmentation in which $\alpha-$expansion with parameter re-estimation performs better than $\alpha-$expansion without parameter re-estimation or ICM. This result was not only promising for textile CAD (Computer-Aided Design) to redesign the textile image, but also for image representation. It was also found that bag of shapes model based on MRF segmentation can obtain comparable classification accuracy with bag of SIFT features in the framework of nearest neighbour class-based matching. Finally, the result indicated that incorporation of graph features extracted by constructing region label graphs can improve the classification accuracy compared to both bag of shapes model and bag of SIFT models.

List of symbols

An effort was made to avoid ambiguities in the notation used in mathematical descriptions. Lowercase Roman letters denote scalars, as in x, except when denoting functions. Lowercase bold Roman letters denote vectors, as in $\mathbf{x}$, and uppercase Roman letters denote constants, as in D. Uppercase bold letters denote matrices, as in $\mathbf{\Sigma}$. Whenever possible, indexing variables use the same letter as the range, as in x_d, $d \in \{1, \ldots, D\}$. The calligraphic $\mathcal{D}$ denotes a data set of observations. Lowercase Greek letters denote model parameter, such as μ.

The following sections list the symbols that are shared across the discussions of different techniques, as well as those that are used in the description of specific techniques.

Shared

$\mathcal{G}$	Graph
$\mathcal{V}$	Vertex set of the graph
$\mathcal{E}$	Edge set of the graph
f_{x_i}	The label of site x_i.
δ	Delta function
$\mathcal{L}$	The label set

Markov Random Field model

$\mathcal{X}$	Discrete set of sites.
$\mathbf{x}_i$	i-th site in $\mathcal{X}$.
f	Label configuration of MRF
$E(f)$	Energy function of MRF label configuration f
f^*	Probable label configuration of MRF
$\mathcal{N}$	Neighbourhood system
$\mathcal{C}$	Cliques
$D_{\mathbf{x}_i}(\cdot)$	Data penalty function
T	Temperature
Z	Partition function
V_c	Clique potential
$U(f)$	Summation of clique potentials
α	label α in $\alpha-$expansion
λ	Coefficient of energy function
k	A cluster index
θ_k	Parameters of k-th Gaussian
μ_k	mean of k-th Gaussian
$\mathbf{\Sigma}_k$	Covariance matrix of k-th Gaussian
K	Number of clusters

Max-flow

$\mathcal{E}_\mathcal{G}$	A cut of a graph $\mathcal{G}$
$\mathcal{S}$,$\mathcal{T}$	Partitions of sets of nodes in graph
s, t	Terminals in a graph
fl	Flow of a graph

$\triangle fl$	The maximum flow pushed that saturates at least one edge in graph
$\mathcal{G}_{fl}$	Residual graph
l	The length of a graph
m	The number of nodes in a graph
n	The number of edges in a graph

$\alpha-$expansion

α, $\bar{\alpha}$	Two terminals of a graph
$\mathcal{P}$	Partition
$\mathbf{x}_a$, $\mathbf{x}_b$	Auxiliary nodes
t, e	Edges
V	Smoothness prior term

Bayesian Information Criteria

$\mathcal{Y}$	The data set
$\hat{\theta}$	Maximum likelihood parameter estimates
$\mathcal{M}_k$	The candidate parametric model which has k classes
$L(\mathcal{M}_k)$	Likelihood of model $\mathcal{M}_k$
N_s	The number of samples
$N_p(\mathcal{M}_k)$	The number of parameters in the model $\mathcal{M}_k$
$BIC(\mathcal{M}_k)$	BIC value of the $k-$th model
k_{BIC}	Minimum BIC value of $k-$th model
d	The dimension of the data

Segmentation evaluation model

M,N	Numbers of labels in segmentations
$R(M,N)$	The number of possible mapping of the N labels in one segmentation to the M labels in the other segmentation
U	Unmatched labels between two segmentations
$\mathcal{U}$	The percentage of pixels with unmatched labels
err	Segmentation error

Bag of shapes model

M	Width of a shape image
N	Height of a shape image
$f(x,y)$	Shape image
(x_c,y_c)	The centroid of a shape image
$f(r,\theta)$	Shape image in polar space
$F(\rho,\phi)$	The polar Fourier spectrum
R	The resolution of radial frequency in the polar space
T	The resolution of angular frequency
r	The radius in polar space of image plane
θ	The angule in polar space of image plane
$\mathbf{d}$	A generic Fourier descriptor
ρ	The radius in polar space of frequency plane
ϕ	The angule in polar space of frequency plane
S	The number of shapes

T	The length of the shape feature vector

Naive Bayes Nearest Neighbour

Q	A query image
C	A class
$\hat{C}$	The optimal class
d_n	$n-$th descriptor
$p(\mathbf{d}\|C)$	The class likelihood function
M	The number of the descriptors
K	Kernel function
K_g	Gaussian kernel
K_e	Exponential kernel

Region label graphs

$G_{adj}(G'_{adj})$	The graph feature obtained from a graph (complement) constructed considering the adjacency of the graph (complement) nodes
$G_{dis}(G'_{dis})$	The graph feature obtained from a graph (complement) constructed considering the dissimilarity of the graph (complement) nodes
$\mathcal{D}$	Subset of vertex set $\mathcal{V}$
v	Each vertex in $\mathcal{V}$
k	Chromatic number

Support Vector Machine classification

C_p	Penalty parameter to penalise the error term
ξ_i	Errors
ϕ	Mapping function
$\mathbf{x}_i$	Training vectors
L	The number of the training vectors
$y_i \in (-1, 1)$	Class label

Chapter 1

Introduction

This PhD research was carried out as a part of the FABRIC (Fashion and Apparel Browsing for Inspirational Content) project. FABRIC was a collaborative project developing innovative technology to improve browsing, retrieval, and management of digital images throughout the textile and clothing industry, cultural heritage, and other sectors that access digital image collections.

This thesis reports work carried out by the author at the School of Computing of the University of Dundee concerning image content understanding and image representation for image classification. This chapter first describes the thesis objectives and motivation, then explains the challenges involved and summarises the contributions. Finally the structure of the thesis is provided.

1.1 Objective

The main goal of this thesis is to contribute novel image representations and extracted features that can be used for image comparison or matching that is important in many applications in computer vision, such as image classification, retrieval, and browsing, etc. The type of image representation we discuss here is

obtained by transforming low-level input (e.g. RGB value of each pixel in image) to a higher level form (e.g. segment an image into a collection of regions, and build up a higher layer based on these regions, such as trees, graphs to represent the structure information.). This kind of image representation is usually called an intermediate image representation.

In this thesis, we contribute a novel image representation based on segmented regions. Therefore, finding suitable representation requires building a proper segmentation model, and determing how to use the segmentation regions to represent images and extract features. We also want to investigate the representation and features to see whether they are effective in application. We apply the representation and features on an image classification application. The images analysed here are textile images, Figure 1.1 shows some examples of textile images. The methods proposed in this thesis could be extended to be used with other kinds of images but this is left for future work.

1.2 Motivation

Textile design analysis is an important task and very useful for imitative design and redesign. In textile industry, textile image segmentation can be used for textile CAD (Computer-Aided Design). It is very time consuming for a CAD designer to edit and create patterns and design. If we can extract the design and patterns from the textile images automatically, store and reuse them (e.g. better efficiency in colour selection, and faster edit based on existing design), it will help the designers improve productivity and reduce the time and product development cost in a considerable manner when comparing with laborious manual work of designing.

The extracted design and patterns from segmentation are useful for image representation, and the corresponding extracted features are able to be considered as

Figure 1.1: Examples of textile designs. (Images courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

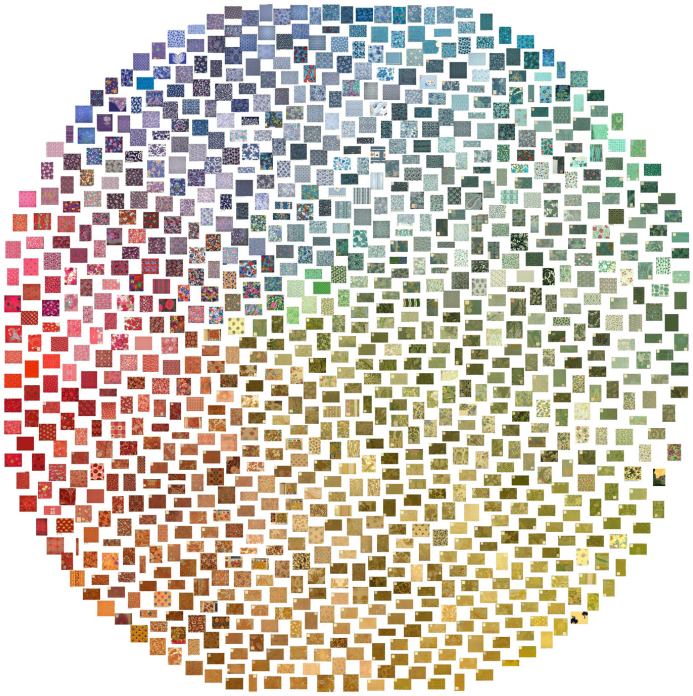
similarity measures to compare and match images in the applications as below:

- Image visualisation and browsing. Figure 1.2(a) shows a 2D image visualisation example. Designers can click any image they are interested and zoom in to look at the details. Images displayed closely have similar features, and they can be similar because of similar regions within images which are segmented. This gives the designers easy access to the images, and potentially fuels their creativity, and improves design efficiency.
- Content based image retrieval (CBIR): In CBIR, users usually give a query example and the system returns the similar searching results (see Figure1.2(b)). Users could point out one region or groups of regions they are interested in within the query example and the system would return the images containing similar regions, and also show where the regions are.

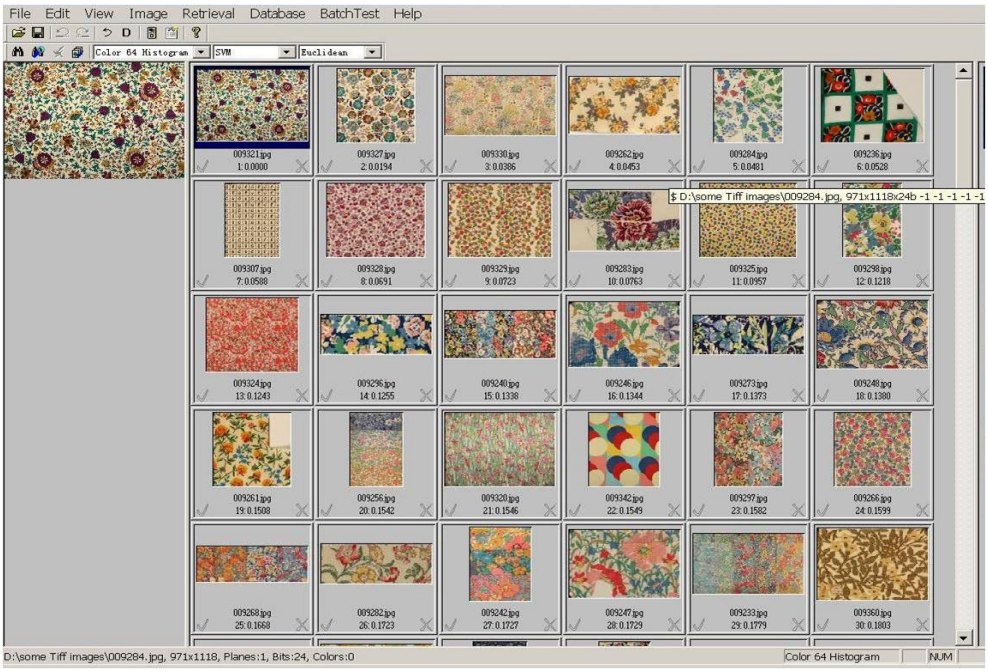
1.3 Challenges

Four reasons why image content understanding and representation are challenging here are:

- Segmentation is an ill-defined problem, and the segmentation ground-truth is not available in Liberty Art Fabric data for the output of an algorithm to compare against.
- Physical texture: Some textile designs have strongly textured physical surfaces. In order to see the texture clearly, small pieces from textile swatches are shown in Figure 1.3. This is affected by the fibres (e.g., smooth or crimped), yarns (e.g., soft or hard twist; number of plies), fabric construc-



(a)



(b)

Figure 1.2: (a)Visualisation of 1000 textile designs using colour features. This figure is produced by Wang *et al.*[129]. (b) An example of image retrieval. (Images courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

Figure 1.3: Some textile swatches with strong texture. (Images courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

tion (e.g., type of weave or knit), and/or textile finishing (e.g., brushing or chemical application). All of these factors may impact the visual texture of the images and increase the difficulties of image representation and feature extraction.

- Images assigned different labels can share similar content: For example, some floral images shown in Figure 1.4(c) are very similar with some leaf images (Figure 1.4(d)), and there are common leaves in the floral images as the leaf images. Also the paisley and leaf (Figure 1.4(a) and Figure 1.4(b)), and check and geometric (Figure 1.4(e) and Figure 1.4(f)) can be easily confused. In both cases they have a very similar appearance and shape. We need a representation that can distinguish images in different classes that have much in common.

- Large intra-class variation: Classes of images do not have precise definitions. There may be many variations among the same kind of images. For example, a floral image can come in many forms. Flowers can have different shapes, as shown in Figure 1.5(a). Also there is not a standard definition of 'geometric', see Figure 1.5(b).

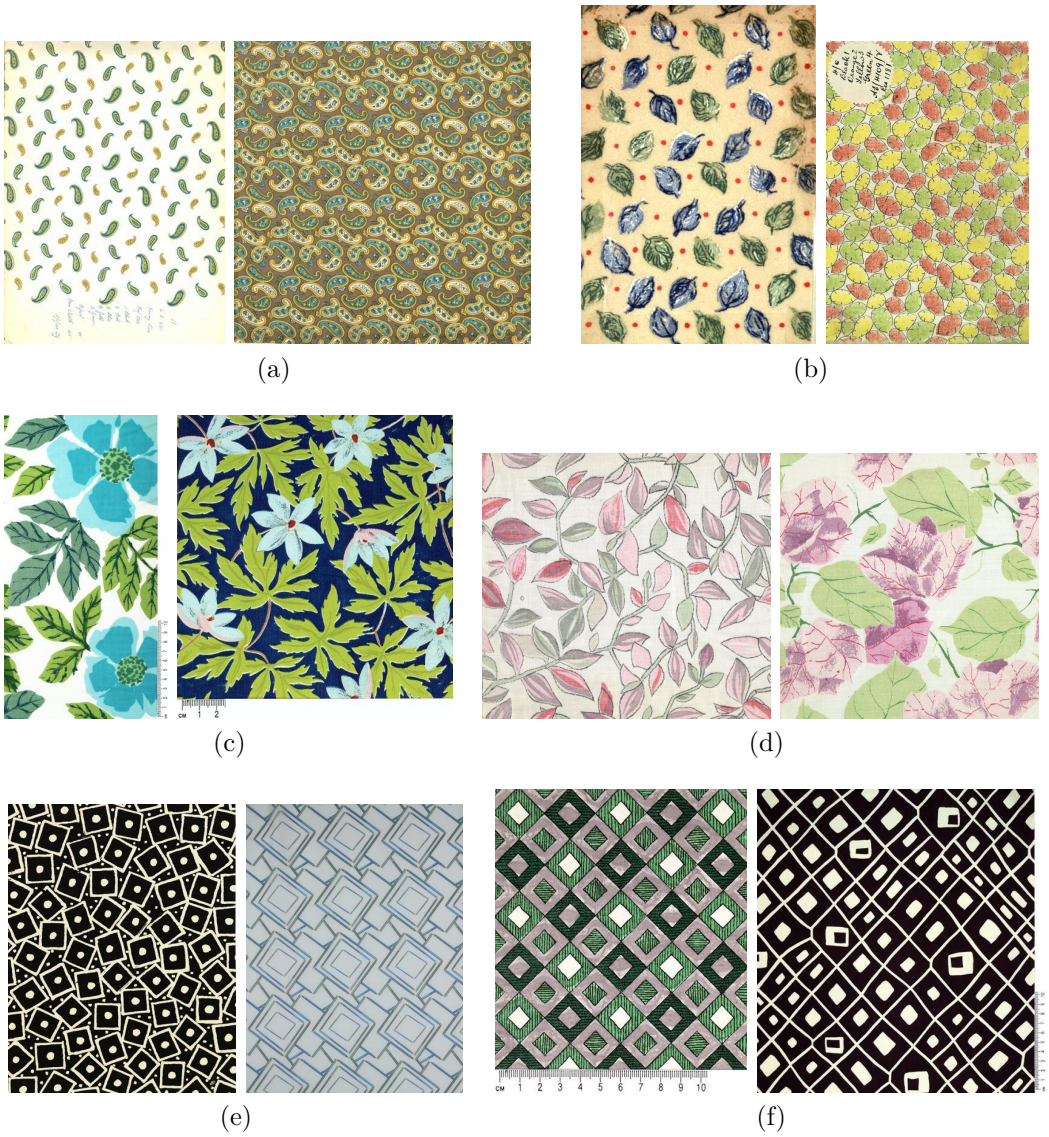


Figure 1.4: Paisley images (a) that could be easily confused with the two leaf images in (b). Floral images (c) have leaves as in leaf images (d). Two geometric images (e) are very similar to the two check images in (f). (Images courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

(a)

(b)

Figure 1.5: (a) Floral images have various forms. (b) Geometric images contain very different shapes. (Images courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

1.4 Contributions

Extracting printed design and woven patterns using MRFs

Novel image representations are proposed based on the extraction of printed design and woven patterns. The problem of extraction is modelled as a pixel labelling problem. This is motivated by printed textile production which assigns distinct colour dyes for different patterns, and prints the dyes on some materials to produce the textile designs. Therefore, segmenting textile images into homogeneous regions sharing similar colours can extract the designs and patterns. However, due to strong physical surface texture, standard clustering methods without considering the spatial coherence fail in design extraction. Markov Random Field (MRF) [19, 17] which can consider the spatial coherence in neighbourhood system was used. This method included: spatial and feature domain image modelling, energy function minimisation, parameter estimation and evaluation. A method for quantitative evaluation is presented to compare the performance of

MRF models. The energy of MRF is optimised using α - expansion and iterated conditional modes (ICM), both with and without parameter re-estimation. The pixel labelling produced suggests ways of compactly representing image content in terms of colour and shape, and is promising for image classification.

Textile classification using Bags of shapes

The bag of words model is a very popular method to represent images in the application of classification and recognition [145, 39, 29]. However, most works use a collection of regions detected using region detectors [91] or patches uniformly obtained from the original image [39] as visual words rather than segmented regions. This is possibly because of segmentation errors. However, since it is shown that reasonable segmented regions can be achieved by using MRF pixel labelling, a novel image representation, bags of shapes, based on MRF segmentation results was proposed. In this model, a collection of segmented regions are simply obtained by extracting connected components from MRF pixel labelling results. Each component is described using a 48 dimensional Generic Fourier Descriptor (GFD) [143]. Images are thus represented as bags of shape descriptors. Unlike the normal bag of words model, there is no need to calculate the codewords using some clustering method, e.g. k - means, to quantise the descriptors and represent images using histograms of codewords. An image represented using bags of shapes model is considered as a matrix of n by 48, where n is the number of regions contained in each image and different images have different number n. This avoids the information loss caused by clustering descriptors. For classification, Naive Bayes Nearest Neighbour [14] classification method is used to investigate the use of bags of shapes model. Classification performance is compared to that obtained with bags of SIFT features [80]. Matching using bags of shapes is considerably faster and of comparable accuracy. However, bags of shapes models ignore the MRF labels, which might be useful in tasks such as classification.

Textile classification using region label graphs

Novel region label graphs are proposed to incorporate spatial information in bags of shapes models for textile classification. As well as partitioning an image into regions, MRF labelling identifies groups of regions that have the same label value. Regions with the same label are treated as a group. Each group is associated uniquely with a vertex in an undirected, weighted graph and is represented as a bag of shape descriptors. Edges in graphs denote either the extent to which the groups' regions are spatially adjacent or the dissimilarity of their respective bags of shapes. Each single shape is described using GFD. Codewords are calculated and shape features of images and groups of regions are histograms of the codewords. With graphs constructed, series of unweighted graphs are obtained by removing edges in order of weight. Fixed dimensional graph feature vectors are calculated from the series of graphs and their complements based on the graph theoretic features: chromatic number and domination number. The weights which cause a change in chromatic number or domination number are defined as graph features. Image features are combinations of shape features and graph features. The representation and combination features are investigated using linear Support Vector Machines (SVM). The classification results show that an improvement in classification can be obtained by combining shape features with the proposed graph feature vectors.

In summary, the main contributions of this thesis are:

- Defining the problem of extraction of printed design and woven patterns, and apply MRF pixel labelling to extract the designs and patterns.
- Based on MRF pixel labelling results, the bags of shapes model was proposed and tested in the application of textile design classification. Comparison between bags of shapes model and bags of SIFT model is made.

- A novel method which uses MRF segmentation labels including graph construction and feature extraction was proposed to capture the spatial information in images. The spatial information was calculated based on chromatic number and domination number. Experimental results showed that using spatial information in this way was helpful in improving image classification accuracy.

1.5 Overview of the system

In this thesis, a system of classification using region label graphs is provided. The pipeline of the system is as follows. An image is segmented to get K label images first, and connected components (regions) are then extracted from every label image. For each region, a Generic Fourier descriptor (GFD) is calculated. A codeword dictionary is built using a clustering methods, e.g. k means. The image is thus represented as a histogram of codewords. To encode spatial information, a graph is constructed from the MRF labelling results. There are two methods of constructing the graph and extracting the graph features: (1) Construct an undirected complete or non complete graph, and label each edge as the arc length of the common boundary shared by groups of regions. In this case, the extraction of graph features need no codewords calculation and feature quantisation. (2) Construct an undirected complete graph, and label each edge as the dissimilarity of a pair of groups of regions. In this case, the extraction of graph features need codewords calculation and feature quantisation. In both methods, each vertex of the graph is associated with a group of regions which share the same label. The graph features are calculated using chromatic number or domination number. Finally, the image is represented as the combination of the shape feature and graph features, and the image feature can be used for any standard classification scheme, e.g. SVM. A flowchart of the system is shown in Figure 1.6.

1.6 Structure of thesis

The thesis is composed of five chapters as below:

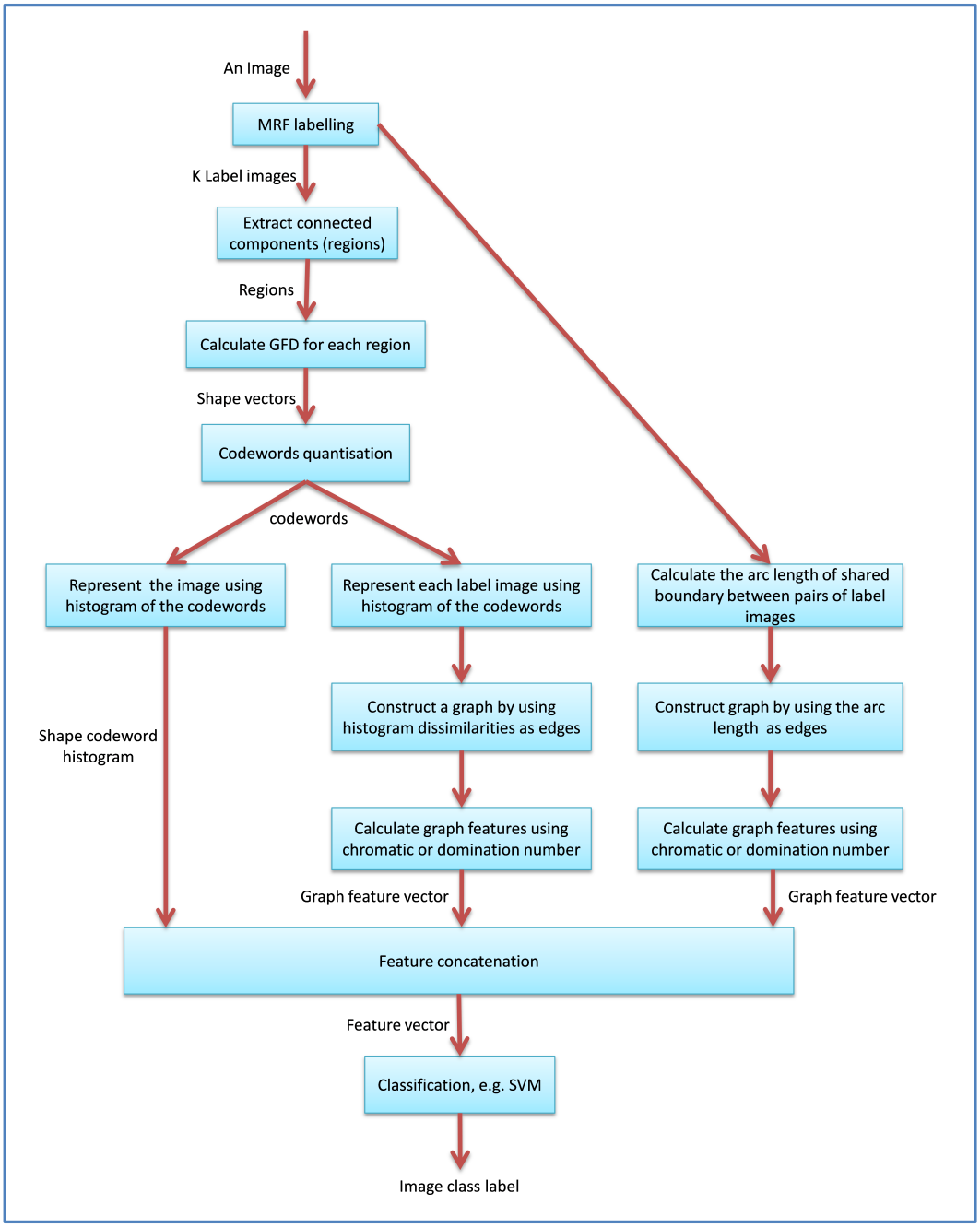


Figure 1.6: System flowchart of classification using region label graphs

Chapter 1 describes the motivation and objectives of this research as well as the main contributions.

Chapter 2 states the textile segmentation problem first and overviews some the related work on segmentation. It then builds up the MRF pixel labelling method for textile design. Finally, it presents the experiments and results, and draws some conclusions.

Chapter 3 begins with a review of literature related to image representation and then proposes novel image representations for textile design. Graph features are obtained by constructing novel region label graphs based on MRF pixel labelling results.

Chapter 4 overviews Nearest-Neighbour based classifier, investigates proposed image representation using this classification method, reports some experiment results, and draws some conclusions. It then sets up experiments to test graph features using linear SVM classifier, reports some results, and draws some conclusions finally.

Chapter 5 summarises the conclusions drawn from the accomplished experiments and discusses possible future directions.

Chapter 2

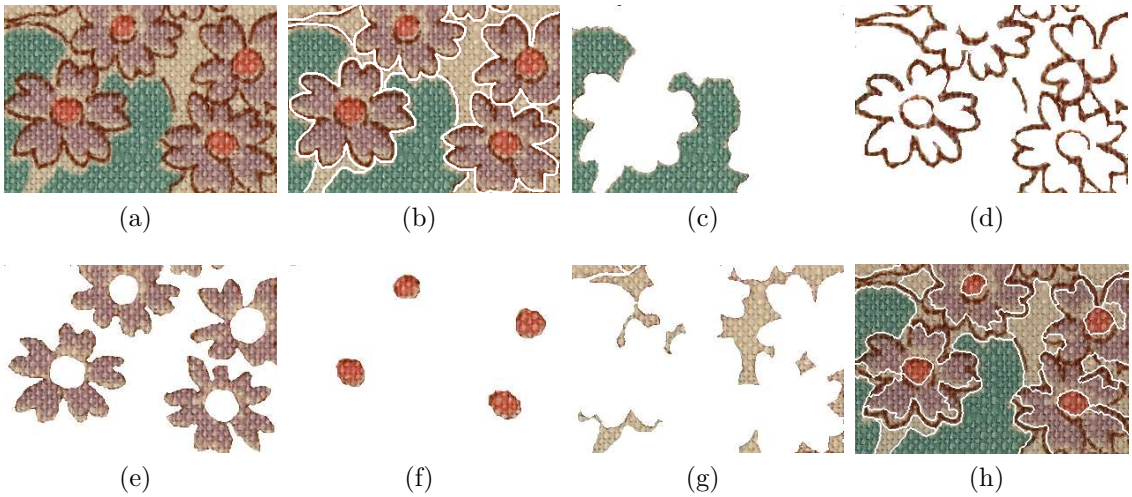
Textile image segmentation using Markov Random Fields

2.1 Introduction

Accurate automatic extraction of printed designs and woven patterns from images of textiles is an important task in providing a suitable representation on which to build algorithms for the classification of digital textile archives. Automatic classification is conducive to the preservation, collection, and management of textile images through the textile industry, and can help designers access to images more easily and view images more efficiently. Besides, pattern extraction is an important textile image processing for imitative textile CAD (Computer-Aided Design) to redesign the textile images.

The extraction task can be considered as a problem of image segmentation. This chapter describes methods for colour segmentation of digital images and evaluates them on a historical commercial archive owned by Liberty Art Fabrics. The segmentation problem is ill-defined with more than one reasonable segmentation solution for a given image. Figure 2.1(a) shows a six-colour design with sketchy

flowers and Figure 2.1(b) illustrates one possible manual segmentation of that design into different design components (e.g., flowers and leaves).



(a) (b) (c) (d)

(e) (f) (g) (h)

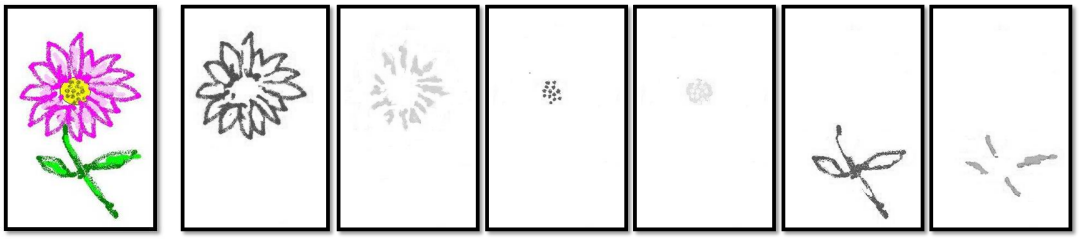
Figure 2.1: (a) A textile image (Images courtesy of Liberty Art Fabric. ©Copyright protected; Reproduction not permitted.) (b) One of many reasonable (manual) segmentations. (c-g) (Manual) segmentation result motivated by colour separation in textile production, a manually obtained class labelling of pixels. (h) JSEG segmentation.

The formulation of the problem of segmentation of textile images requires consideration of textile production. There are various methods of textile production, but images from the Liberty Art Fabrics archive are mainly of textiles produced by two kinds of methods:

- Printing: Textile printing is a process of applying colour to fabric in definite patterns or designs. Contemporary production of Liberty Art Fabrics prints requires making a screen for each colour used in a design. This can be done by hand tracing or by computer reduction and separation. Figure 2.2 shows a simple example of the colour separation procedure of a design. Dye is applied to the fabric by flat screens, e.g., engraved plates (see Figure 2.3(a)), for short runs, and rotary screens, e.g., rollers (see Figure 2.3(b)), for long runs and continuous designs.

(a) (b)

Figure 2.3: Textile printing. (a) Engraving. (b) Roller printing. (Images are from Anna Buruma)

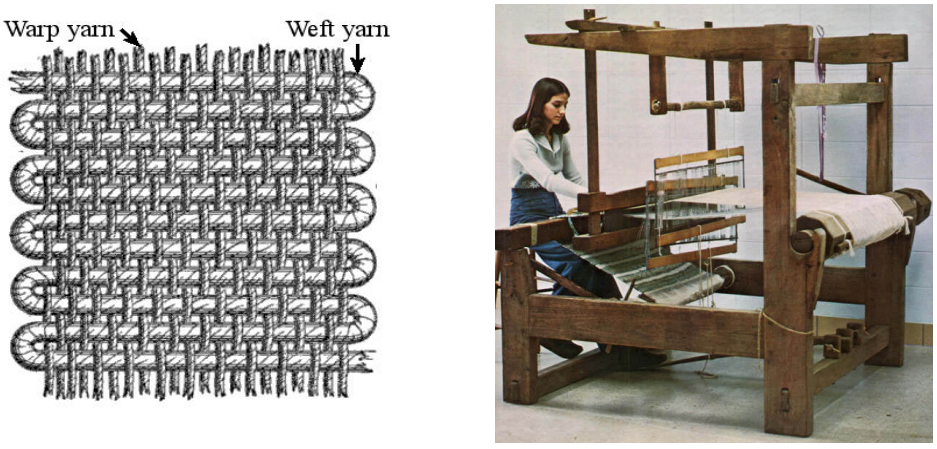


(a) (b)

Figure 2.2: Colour separation of a design. (a) A design. (b) Colour separation. (Images are from Anna Buruma)

- Weaving : In general, basic weaving process involves the interlacing of two distinct sets of yarns or threads at right angles to each other: the warp and the weft (see Figure 2.4(a)). The warp yarns run lengthways and the weft runs across from side to side. Fabric is woven on a loom, a device that holds wrap yarns taut and in parallel order while weft yarns are woven through them (see Figure 2.4(b)).

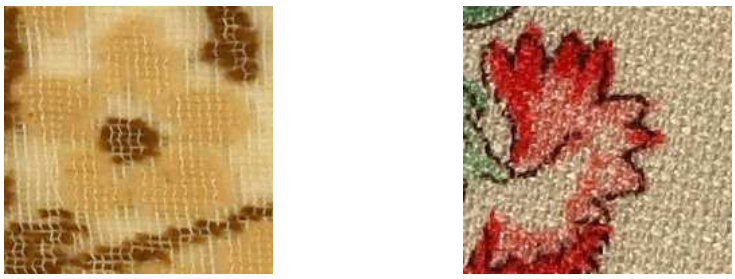
In Liberty Art Fabrics archive dataset, some designs are printed on the surface of the fabric through the printing process and other designs are woven into the fabric through the weaving process. Although many of the images are from the previous 50 years, there are many that are much older, are only fragments, or have degraded over time.



(a) (b)

Figure 2.4: (a) Warp and weft in plain weaving. (b) Weaving on a loom. (Images are from Wikimedia Commons)

Motivated by the process of colour separation in textile production, textile images are reasonably segmented into homogeneous regions sharing the similar colour. Therefore, the segmentation can be defined as a multi-class pixel labelling problem. Every pixel in an image is assigned a label, and pixels with similar colour share the same label. The number of labels is the number of colour dyes used in the textiles. Figure 2.1 (c-g) shows a ground truth for the original image in Figure 2.1(a) that was obtained by manually assigning its pixels to one of five classes (i.e., colours). However, in a large database, manual labelling is not feasible. Thus, the problem to be solved is to determine the number of classes (colours), in a digital textile image and label each pixel as belonging to one of these classes. Solutions to this problem can be quantitatively evaluated by comparing computer extractions with manually-annotated ground-truths such as those shown in Figure 2.1 (b-g).



(a) (b)

Figure 2.5: (a) The underlying fabric can be strongly textured. (b) Colour dyes can blend together and have spatially varying density. (Images courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

Another aspect that may complicate the problem of segmenting a digital image into colour regions is the physical surface texture of the textile images (see Figure 2.5). Further difficulties arise due to the age and condition of the textiles, especially in historic archives. For printed textiles, the saturation of a dye varying within a region (either due to degradation or by design) may be problematic. Printing may result in screens being misaligned and in dyes blending together near their boundaries to form new colours. However, these effects are sometimes deliberately introduced to create additional colour variation or to help ensure that the fabric is entirely covered with dye leaving no undyed areas.

Markov random field (MRF) models have been used previously for image segmentation, though usually for binary foreground-background segmentation tasks. In this thesis, an MRF model was formulated and applied to the textile segmentation task described above. Given ground-truth estimates of colour separation, this application provided a useful test case for comparing algorithms. A method for quantitative evaluation of multi-class pixel labelling methods was used for segmentations with different numbers of labels. It was used to compare segmentation results obtained using Gaussian mixture colour clustering, MRF inference using Iterated Conditional Modes (ICM), and MRF inference using α - expansion. The MRF models considered both the class colour distributions and the spatial coherence of the pixel labels. The colour distributions were learned based

on an initial labelling estimated using a Gaussian mixture in colour space and the number of colour classes was estimated using the Bayes Information Criterion (BIC).

The remainder of this chapter is organised as follows. Section 2.2 introduces some related work on image segmentation. Section 2.3 provides an overview of Markov Random Field pixel labelling, including minimisation of MRF energy functions, model selection, and parameter estimation. Section 2.4 proposes a novel quantitative segmentation evaluation method. Section 2.5 provides details of the actual experiments carried out and the results. Finally, Section 2.6 discusses the results and presents some conclusions.

2.2 Related work

Image segmentation has long been an important and challenging topic. It aims to partition an image into several disjoint regions that are homogeneous with regards to some measures, often as a pre-treatment to higher level computer vision processing. Feature space clustering is a popular approach in image segmentation, in which a feature vector of local properties is computed at each pixel. The feature space is then classified into different clusters using clustering methods [57, 135] and each pixel is assigned a label based on the cluster that contains its feature vector.

In computer vision, much research has been done based on feature space clustering [22, 26, 27, 133, 99, 35]. Some authors use Gaussian mixture models (GMM) [133, 99], in which each component describes a cluster in the feature space and has Gaussian distribution. The associated parameters can be estimated through maximum likelihood (ML) estimation using an Expectation-Maximisation (EM) algorithm [33, 121, 10]. The drawback of this approach to segmentation is that

the commonality of location is not taken into account when grouping the data. In other words, the prior knowledge that adjacent pixels most likely belong to the same cluster is not used, so the results often lack spatial coherence, especially when the images exhibit strong texture. Images of fabric have been segmented using colour clustering after first attenuating noise and texture by applying median filtering [124]. However, the use of a median filter as a pre-prosessing step can result in important details being lost. The JSEG segmentation algorithm [35] performs colour clustering at the pixel level and then uses region growing on the cluster labels to segment colour-texture regions. An example JSEG result is shown in Figure 2.1(h) using the default parameter settings [34]. Whilst it can find reasonable colour-texture regions, it has two drawbacks. Firstly, it decomposes the task into two steps of colour clustering and region finding so that the colour clustering step takes no account of spatial information. Secondly, fabric designs have spatially disjoint regions that share the same colour-texture. We seek a labelling that makes this explicit.

Imposing spatial smoothness is a key element to many image applications since it is a significant *a priori* property of images [11]. In Bayesian approaches, spatial coherence is usually imposed by a Markov random field (MRF) prior [78]. Many works have been reported to show that MRF is a very powerful framework to design efficient and robust models for image segmentation [83, 84, 104, 62, 140]. These approaches have used maximum a posteriori (MAP) estimation in conjunction with MRF, the MAP-MRF framework. In this framework, the objective is the joint posterior probability of the MRF labels. The form and parameters are determined according to Bayes rule, by those of the prior of the labels and the conditional probability of the observed data. MRF modelling is to derive the form of the posterior distribution and to estimate the parameters in it so as to define the posterior probability completely. Another important part is to maximise the posterior probability. The MAP solution is equivalently found by minimising the posterior energy function.

The energy function of an MRF usually consists of two terms: a data cost penalising the disagreement of the data and the label, and a smoothness cost enforcing spatial coherence. Early approaches, such as Iterated Conditional Modes (ICM) [8] or simulated annealing [6] can be ineffective at minimising the energy function. However, over the last few years, powerful new optimisation algorithms such as graph cuts [65, 20, 18] and Loopy Belief Propagation (LBP) [138] have emerged to become very popular optimisation methods. Szeliski *et al.* [117] have compared six MRF optimisation methods on various computer vision problems. They show that methods based on graph cuts using the max-flow algorithm [46] and LBP are good solutions, and α - expansion, one of the graph cuts algorithms is proved to be very effective. For non-binary segmentation, such as multi-labelling problem, max-flow is iteratively used in α - expansion to reach an optimum. α - expansion is guaranteed to converge and is effective in practice. In recent work, *Delong et al.* [31] extend α - expansion to minimise the energy function with label cost that can penalise a solution based on the set of labels, i.e. penalising the use of too many labels. The label costs may be useful when labels belong to known classes with specific priors (The implementation is available at [127]). Another method to obtain the MAP estimates of MRF is max-sum, which is based on an efficient linear programming solver [88], and allows multi-labelling segmentation to be done directly in a global manner [130]. There are other methods e.g. second order core programing relaxation [69], and sequential tree-reweighted max-product message passing [64]. However, the problem of solving the multi-label MAP of the MRF is NP-hard. Various approximations are taken into account to reach a good suboptimal solution.

2.3 Markov Random Field labelling model

A *labelling problem* is specified in terms of a set of *sites* (e.g. pixels, features, data points) and a set of *labels* (e.g. categories, geometric models, disparities).

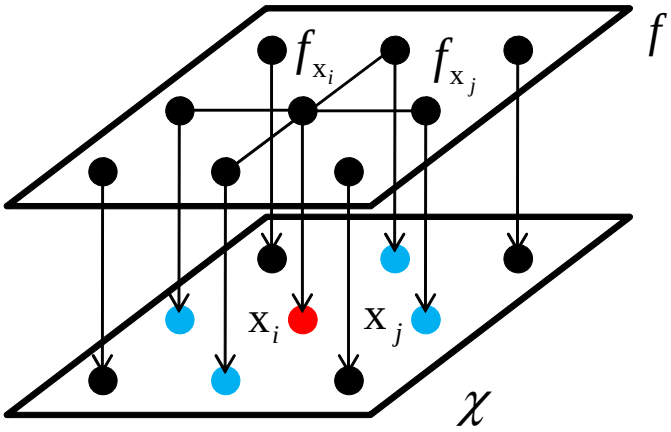


Figure 2.6: Markov Random Field model

Let $\mathcal{X}$ denote a set of discrete sites:

$$\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_i\}$$

and let $\mathcal{L}$ denote a set of discrete labels:

$$\mathcal{L} = \{1, ..., \mathrm{K}\}$$

Labelling is to assign a label from the label set to each of the sites $\mathbf{x}_i$ in $\mathcal{X}$. The sites in $\mathcal{X}$ are related to one another via a neighbourhood system. The neighbourhood system of $\mathcal{X}$ is:

$$\mathcal{N} = \{\mathcal{N}_i | \forall \mathbf{x}_i \in \mathcal{X}\}$$

An MRF is always defined with respect to a neigbourhood system. In MRF's, only neighbouring labels have direct interactions with each other. In the terminology of random fields, a labelling is called a *configuration.* The purpose is to find the labelling f^* that is most probable given $\mathcal{X}$.

$$f^* = argmax\,[P(f|\mathcal{X})] \tag{2.1}$$

This is known as the MAP estimate. According to Bayes' rule, the posterior probability is:

$$P(f|\mathcal{X}) = \frac{P(\mathcal{X}|f)P(f)}{P(\mathcal{X})} \tag{2.2}$$

where $P(f)$ is the prior probability of labellings f, and $P(\mathcal{X}|f)$ is the conditional probability density function of the observations $\mathcal{X}$, also called the likelihood function of f for fixed $\mathcal{X}$. $P(\mathcal{X})$ is a constant when $\mathcal{X}$ is given. Therefore

$$P(f|\mathcal{X}) \propto P(\mathcal{X}|f)P(f) \tag{2.3}$$

MAP estimation is equivalent to finding

$$f^* = argmax[P(\mathcal{X}|f)P(f)] = argmin[-(\ln P(\mathcal{X}|f) + \ln P(f))] \tag{2.4}$$

and the energy function can be defined as:

$$E(f) = -\ln P(\mathcal{X}|f) - \ln P(f) \tag{2.5}$$

The task is to minimise the energy function.

To define an MRF in terms of a Gibbs distribution, cliques need to be introduced. A clique is a subset of sites in the neighbourhood system. In the first order neighbourhood system, also called the 4-neighbourhood system, every site has four neighbours, as shown in Figure 2.7(b). In the second-order neighbourhood system, also called the 8-neighbourhood system, every site has eight neighbours, as shown in Figure 2.7(d). Figure 2.7(e) show clique types for the first and second order neighbourhood systems for the lattice. The types of cliques in first order neighbourhood system are the single site and the horizontal and vertical pair-site cliques. The second order neighbourhood system includes all of the clique types shown in 2.7(e): single site, pair-site, triple-site and quadruple-site cliques.

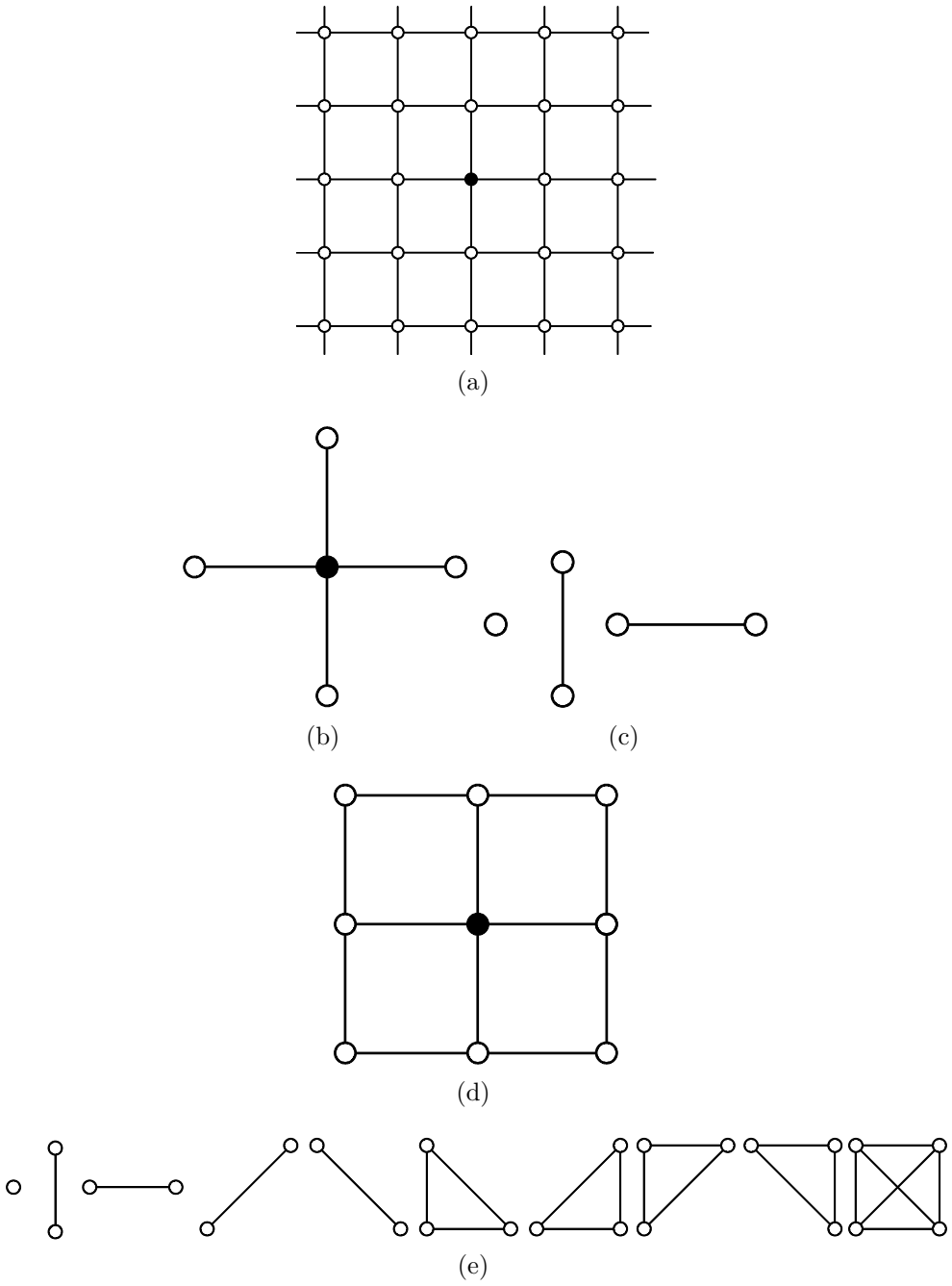


Figure 2.7: (a) A lattice. (b) A site (filled circle) and its neighbours (unfilled circles) used in first-order neighbourhood system. (c) Cliques on a lattice of a regular site in first-order system. (d) A site (filled circle) and its neighbours (unfilled circles) used in second-order neighbourhood system. (e) Cliques on a lattice of a regular site in second-order neighbourhood system.

Hammersley and Clifford [48] proved the equivalence between Markov random fields and Gibbs distributions. This provides a mathematically tractable means of specifying the prior. A *Gibbs distribution* takes the form:

$$P(f) = \frac{1}{Z} \times \exp(-\frac{1}{T}U(f)) \tag{2.6}$$

where $Z = \sum \exp(-\frac{1}{T}U(f))$ is a normalising constant called the *partition function* which is a sum over all possible configurations. T is a constant called the *temperature* which will be assumed to be 1 unless otherwise stated, and

$$U(f) = \sum_{c \in \mathcal{C}} V_c(f) \tag{2.7}$$

is the *energy function*. It is a summation of *clique potentials* $V_c(f)$ over all possible cliques $\mathcal{C}$. $V_c(f)$ is dependent on the local configuration on each clique c. The MRF used in this thesis is the first-order neighbourhood system and the clique potentials involve vertical and horizontal pair-site [19], so that:

$$P(f) \propto \exp(-\sum_{c \in \mathcal{C}} V_c(f)) = \exp(-\sum_{\mathbf{x}_i} \sum_{\mathbf{x}_j} V(f_{\mathbf{x}_i}, f_{\mathbf{x}_j})) \tag{2.8}$$

An important discontinuity preserving metric function is given by the Potts model [101]:

$$V(f_{\mathbf{x}_i}, f_{\mathbf{x}_j}) = \lambda \cdot (1 - \delta(f_{\mathbf{x}_i} - f_{\mathbf{x}_j})) \tag{2.9}$$

The Potts model has a clique potential for any pair of neighbouring sites $\mathbf{x}_i$ and $\mathbf{x}_j$. $\delta(\cdot)$ represents the Kroneker delta function. If labels of $\mathbf{x}_i$ and $\mathbf{x}_j$ are different ($f_{\mathbf{x}_i} \neq f_{\mathbf{x}_j}$), then its value is λ. Otherwise its value is zero. The constant $\lambda \geq 0$ is a coefficient that specifies the penalty for assigning different labels to neighbouring pixels.

Note that $\mathcal{X}$ are pixel features, and are assumed independent given the labels, so

$$P(\mathcal{X}|f) = \prod_i P(\mathbf{x}_i|f_{\mathbf{x}_i}) \tag{2.10}$$

The likelihood of a label, $f_{\mathbf{x}_i}$, given one of the sites $\mathbf{x}_i$, can be modelled as Gaussian,

$$P(\mathbf{x}_i|f_{\mathbf{x}_i} = k) = \frac{\exp[-\frac{1}{2}(\mathbf{x}_i - \mu_k)^T \mathbf{\Sigma}_k^{-1}(\mathbf{x}_i - \mu_k)]}{\sqrt{(2\pi)^d |\mathbf{\Sigma}_k|}} \tag{2.11}$$

with parameter set θ_k which consists of the mean, μ_{k}, and the covariance matrix, $\mathbf{\Sigma}_k$.

Combining Equations (2.5) by Equation (2.9), (2.10), the energy function of MRF becomes:

$$E(f) = -\sum_{\mathbf{x}_i} \ln(P(\mathbf{x}_i|f_{\mathbf{x}_i}) + \sum_{\mathbf{x}_i}\sum_{\mathbf{x}_j} \lambda \cdot (1 - \delta(f_{\mathbf{x}_i} - f_{\mathbf{x}_j})) \tag{2.12}$$

The first term is the data penalty function, It typically indicates individual label-preferences of pixels based on observed features and pre-specified likelihood function. The second term specifies the penalty for assigning different labels to neighbouring pixels.

2.3.1 Energy minimisation algorithm

How to minimise the energy function is one major part of an MRF model. Many publications in computer vision use graph-based energy minimisation techniques for image segmentation [55, 17]. The energy function addressed by Greig *et al.* [46], and most graph based methods (eg. [107, 126]) takes the form of (2.7).

2.3.1.1 Graph cuts

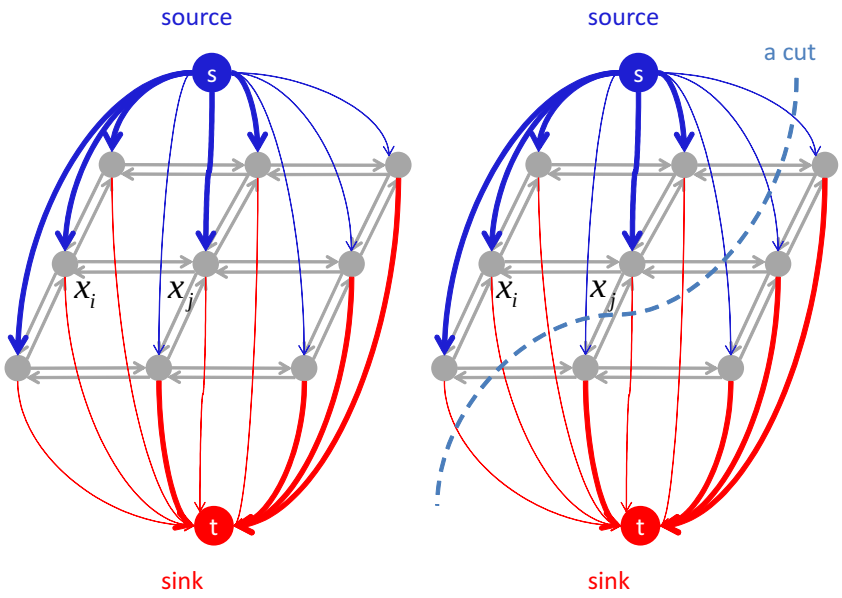
Graph cuts is a type of operation on a graph that use the max-flow algorithm to solve discrete energy function minimisation problems [78]. Greig *et al.* [46] first proposed graph cuts to solve the global minimisation problem of a two-label MRF model (Ising model), Roy and Cox [106] extended it to solve convex multi-label problems, and Boykov *et al.* [20] then used it to approximate the global minimisation for more general multi-label MRF problem. Graph cuts has now been widely used in computer vision and image analysis problems such as image segmentation, restoration, and stereo.

Greig *et al.* [46] were the first to introduce the max-flow algorithm from combinatorial optimisation to minimise certain important energy functions in vision. Max-Flow, also called min-cut or $s-t$ cut, plays an important role in graph cuts. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph with nonnegative weight on each edge, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ the set of edges. In max-flow, $\mathcal{V}$ contains two special vertices, the source s and sink t (see Figure 2.8(a)). A cut $\mathcal{E}_c$ of $\mathcal{G}$ is a subset of $\mathcal{E}$ and satisfies that (1) the subgraph excluding $\mathcal{E}_c$ is disconnected and (2) adding any edges in $\mathcal{E}_c$ to the subgraph, the subgraph is connected. The cost of a cut $\mathcal{E}_c$ is the summation of its edge weights.

There are two kinds of edges: $t-$link and $n-$link. All the edges of the graph are assigned weights or costs. $t-$links connect pixels with terminals. The cost of a $t-$link corresponds to the penalty for assigning the labels to the pixels. $n-$links connect pairs of pixels. The cost of an n-link corresponds to the penalty for discontinuity between the pixels. An $s-t$ cut on a graph with two terminals is a partitioning of nodes in the graph into two disjoint subsets $\mathcal{S}$ and $\mathcal{T}$, $s \in \mathcal{S}$, $t \in \mathcal{T}$. Figure 2.8(b) shows a cut. The min-cut problem is to find the cut that has the minimum cost. According to Ford-Fulkerson theorem [41], this problem is equivalent to finding the maximum flow from s to t.

In combinatorial optimisation, there are many polynomial algorithms for max-flow, and most of the algorithms belong to one of the two following groups: Ford-Fulkerson style augmenting paths [41] and Goldberg-Tarjan style push-label methods [44]. A comparison of these algorithms for energy minimisation in image analysis can be found in [18].

Standard augmenting paths based algorithms, such as the Dinic algorithm [36], work by pushing flow through non-saturated paths from s to t until the maximum flow in graph $\mathcal{G}$ is reached. They store the information about flow fl from s to t in a residual graph $\mathcal{G}_{fl}$. The capacity of an edge in $\mathcal{G}_{fl}$ reflects the residual capacity of the edge in graph $\mathcal{G}$. The graph is initialised with flow $fl = 0$, and edge capacities of residual graph are the same as the original capacities in $\mathcal{G}$. In each iteration, the algorithm finds the shortest path from s to t along non-saturated edges in the residual graph. Once a path is found, the possible maximum flow Δfl is pushed that saturates at least one edge in the path. This is called one augmentation. So the residual capacities of edges are reduced by Δfl and the residual capacities of the reverse edges are increased by Δfl. After one augmentation, the total flow from s to t is increased as $fl = fl + \Delta fl$. The maximum flow is reached when there is at least one saturated edge in any of the path from s to t in the residual graph $\mathcal{G}_{fl}$. The Dinic algorithm uses breadth-first search to find the shortest paths from s to t on the residual graph $\mathcal{G}_{fl}$. After all the shortest paths of a fixed length l are saturated, it starts the breadth-first search of $l + 1$ length from s to t from scratch. Searching for the shortest path is a key factor that improves the time complexity based on augmenting paths. The worst case running time complexity is $O(m(n)^2)$ where n is the number of nodes and m is the number of edges in the graph.



(a) (b)

Figure 2.8: (a) A source and sink graph. (b) An example s-t cut of graph (a)

Boykov and Kolmogorov [18] present a new algorithm based on the standard augmenting path. Similarly to the Dinic algorithm, it finds augmentation path by creating a search tree. However it builds two search trees $\mathcal{S}$ and $\mathcal{T}$, one from s ($s \in S$) and the other from t ($t \in \mathcal{T}$) (see Figure 2.9). Another difference is that these trees are reused and never recreated from zero. The disadvantage of this method is that no shortest augmenting path is guaranteed. Time complexity, technically speaking, is worse than the standard one. However, comparison experiments show that the variant of max-flow algorithm outperforms the standard method, and is faster in many cases [18]. This algorithm consists of three stages: growth stage, augmentation stage and adoption stage.

In the growth stage, search trees $\mathcal{S}$ and $\mathcal{T}$ grow until they touch and a path from s to t is detected. The active nodes (circles noted as "A" in Figure 2.9) find non-saturated edges and make the free nodes (black circles in Figure 2.9) become their children and new active nodes along non-saturated edges. Once all the neighbours of given active nodes become active, they become passive nodes noted as "P" in Figure 2.9. The termination condition is that an active node

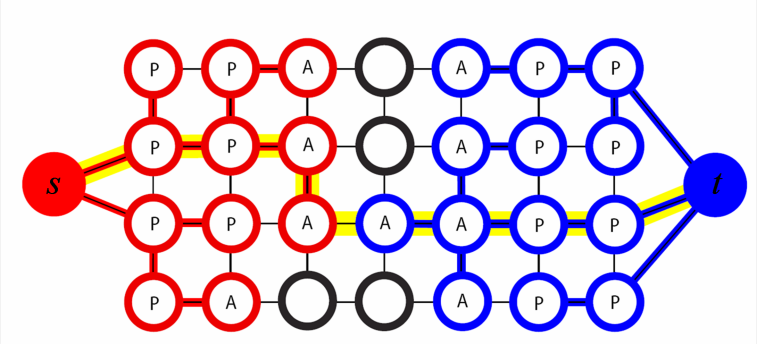


Figure 2.9: Search trees of variant algorithm of max-flow (Boykov and Kolmogorov [18]).

encounters a node that belong to the opposite tree. In this case, the yellow path as shown in Figure 2.9 from s to t is detected.

The augmentation stage augments the path found at the growth stage. Since maximum flow is pushed through the path, some edges in the path become saturated. Some notes in trees $\mathcal{S}$ or $\mathcal{T}$ become orphans since the edges linking these notes to their parents become saturated. This stage may split search trees into forests, s and t are still roots of search trees and the orphan nodes are roots of other trees.

The purpose at this stage is to find a new parent for each orphan. Parent and orphans should belong to the same tree, $\mathcal{S}$ or $\mathcal{T}$, and be connected through non-saturated edges. Orphans will be removed from S or T and turned into free nodes if there are no qualified parents. Their children become orphans. The stage terminates as long as no orphans left. After the adoption stage, it begins the growth stage again until there is no active node and two trees are split by the saturated edges. Thus, the maximum flow is achieved and the minimum cut

is obtained by $\mathcal{S}$ and $\mathcal{T}$.

2.3.1.2 Multi-label graph cuts

Minimisation of an energy function with multiple discrete labels is an NP-hard problem. Many algorithms have been proposed to approximate the global solution with a lower complexity. Two popular graph cuts algorithms are the swap move algorithm and the $\alpha-$expansion move algorithm [20]. Both algorithms work by iteratively computing the global minimum of a binary labelling problem using binary max-flow algorithm, and they converge quickly. Szeliski *et al.* [117] show that $\alpha-$expansion algorithm is very effective at solving the problem of optimisation. The variant max-flow algorithm described in Section 2.3.1.1 is used for $\alpha-$expansion to find expansion moves in this thesis.

The $\alpha-$expansion algorithm assumes that the smoothness prior term V is a metric:

- $V(f_{\mathbf{x}_i}, f_{\mathbf{x}_j}) \geq 0, V(f_{\mathbf{x}_i}, f_{\mathbf{x}_j}) = 0 \Leftrightarrow f_{\mathbf{x}_i} = f_{\mathbf{x}_j}$,
- $V(f_{\mathbf{x}_i}, f_{\mathbf{x}_j}) = V(f_{\mathbf{x}_j}, f_{\mathbf{x}_i})$,
- $V(f_{\mathbf{x}_i}, f_{\mathbf{x}_j}) \leq V(f_{\mathbf{x}_i}, f_{\mathbf{x}_t}) + V(f_{\mathbf{x}_t}, f_{\mathbf{x}_j})$

$\alpha-$expansion can be considered as iterative $s-t$ cut (Max-Flow). The variable α takes values in $\mathcal{L} = \{1, 2, ..., \mathrm{K}\}$ iteratively. Each iteration considers an $\alpha \in \mathcal{L}$ value and the two-label set of $\{\alpha, \bar{\alpha}\}$. Within one α - expansion, some of the labels are simultaneously changed to α while all others remain unchanged. This gives a new labelling f which is accepted only if it reduces the energy. In general there

are an exponential number of possible expansion moves. A Max-flow algorithm is used iteratively to find the optimal expansion move within one $\alpha-$ expansion. The $\alpha-$expansion algorithm is guaranteed to converge, terminating when there is no energy decrease for all values of α.

For each $\alpha-$expansion, the graph is constructed according to the current label and the different label α. Figure 2.10 shows a 1D graph example for a given α. In this graph, there are two terminals: α and $\overline{\alpha}$, and a set of labels. The current partition $\mathcal{P}$ is: $\mathcal{P}_1 = \{\mathbf{x}_i\}$, $\mathcal{P}_2 = \{\mathbf{x}_j, \mathbf{x}_r\}$, and $\mathcal{P}_\alpha = \{\mathbf{x}_q\}$. For each pair of neighbouring sites, if they have different labels, such as $\mathbf{x}_i$ and $\mathbf{x}_j$, $\mathbf{x}_r$ and $\mathbf{x}_q$, two auxiliary nodes circle $\mathbf{x}_a$ and $\mathbf{x}_b$ are created. The weights assigned to the graph are shown in Table 2.1.

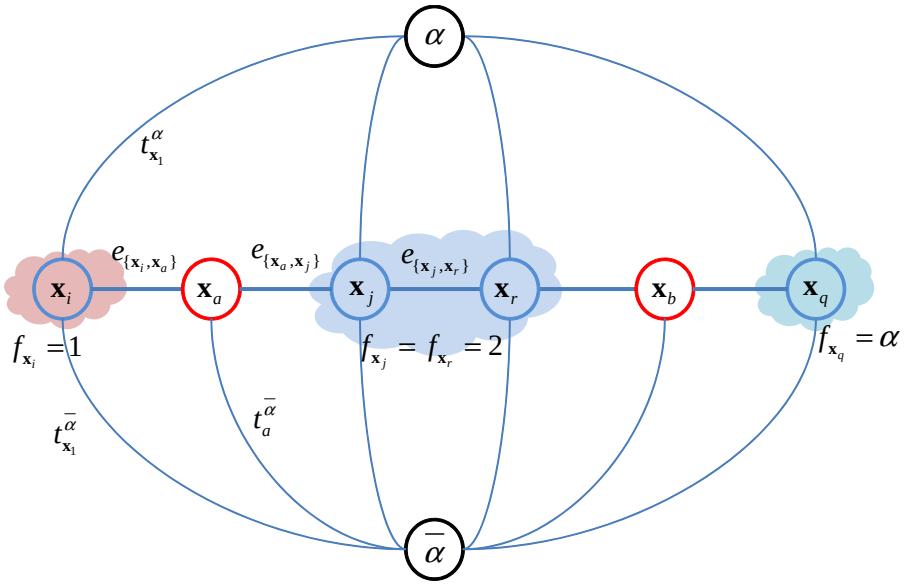


Figure 2.10: Graph construction for a 1D example

edge	weight	
$t^{\overline{\alpha}}_{\mathbf{x}_i}$	$D_{\mathbf{x}_i}(f_{\mathbf{x}_i})$	$f_{\mathbf{x}_i} \neq \alpha$
$t^{\overline{\alpha}}_{\mathbf{x}_q}$	∞	$f_{\mathbf{x}_q} = \alpha$
$t^{\alpha}_{\mathbf{x}_i}$	$D_{\mathbf{x}_i}(\alpha)$	$f_{\mathbf{x}_i} \in \mathcal{K}$
$e_{\{\mathbf{x}_i,\mathbf{x}_a\}}$	$V(f_{\mathbf{x}_i}, \alpha)$	$\{\mathbf{x}_i, \mathbf{x}_j\} \in \mathcal{N}, f_{\mathbf{x}_i} \neq f_{\mathbf{x}_j}$
$e_{\{\mathbf{x}_a,\mathbf{x}_j\}}$	$V(\alpha, f_{\mathbf{x}_j})$	
$t^{\overline{\alpha}}_{a}$	$V(f_{\mathbf{x}_i}, f_{\mathbf{x}_j})$	
$e_{\{\mathbf{x}_j,\mathbf{x}_r\}}$	$V(f_{\mathbf{x}_j}, \alpha)$	$\{\mathbf{x}_j, \mathbf{x}_r\} \in \mathcal{N}, f_{\mathbf{x}_j} \neq f_{\mathbf{x}_r}$

Table 2.1: Weights assigned to the graph

2.3.1.3 Iterated conditional modes

An alternative and older local method, iterated conditional modes (ICM), uses greedy search to find a local minimum [8]. It starts with an estimated label, and for each pixel in turn, the algorithm chooses the label which makes the energy function decrease farthest. This process is repeated until convergence. It is guaranteed to converge and is very fast in practise. However, ICM has been reported to be very sensitive to the initialisation of the labelling.

2.3.2 Parameter estimation

In energy function minimisation, a useful strategy is to alternate energy reduction and parameter estimation in a manner similar to Expectation-Maximisation (EM) [33]. First, the parameters of the clusters should be initialised, and the most popular choice for this is k means. In this thesis, the MRF labels and the parameters of each cluster are initialised by using a GMM, because the model selection is based on this model as well, and actually α - expansion is not very sensitive to the initialisation. In the "E" step, given the fixed Gaussian parameters, a new labelling is found by using α - expansion. In the "M" step, given the new labelling, the parameters of the Gaussians are re-estimated conditioned on this labelling. This results in a new energy function, the value of which the next "E" step should reduce. This is an iterative procedure, and will converge within a small number of iterations. Since parameter estimation is relatively fast, it seems reasonable in the case of $\alpha-$expansion to re-estimate the Gaussian parameters after each new expansion step [140].

2.3.3 The number of segmentation labels

Another issue is how to choose the number of the labels. It is a nagging problem and there is no agreed solution. Due to the difficulty of estimating the number of pixel classes or clusters, in the previous discussions, the number of the labels is assumed to be known. But in completely unsupervised segmentation, this number is unknown.

Some methods (e.g. [132, 73]) in the literature consider unsupervised segmentation as incomplete data problem, in which the image data is observable, but the labels are missing. The associated class model parameters including the number of classes need to be estimated. The model parameters can be estimated via an iterative algorithm in which labels are sampled based on the current estimates of the parameters. The maximum likelihood of the parameters are then computed using the current labelling. The resulting estimates are applied to some model fitting criteria to get the optimal number of labels. There are many criteria in the literature. Akaike Information Criterion (AIC) [2] and Bayesian information criterion (or Schwarz Criterion) (BIC) [109] are increasingly being used for model selection problems. Essentially, these two criteria are based on two different model selection approaches. AIC is aimed at finding the best approximating model to the unknown data generating process while BIC is designed to identify the true model. AIC does not depend directly on sample size. Although BIC takes a similar form to AIC, it is derived within a Bayesian framework, and reflects sample size. For a reasonable sample size, BIC applies a larger penalty than AIC. From a Bayesian view point this motivates the adoption of the Bayesian information criteria. Kato [61] used Reversible Jump Markov Chain Monte Carlo (RJMCMC) proposed by Green [45] for determination of the number of MRF labels. They represented pixel classes by multivariate Gaussian distributions. The number of classes, class model parameters, and pixel labels are all directly sampled from the posterior distribution using an RJMCMC sampler. Kato [61] has shown that a

reasonable number of labels and good segmentation results can be obtained, but comparing with the criteria mentioned, this method is complicated. In my thesis, BIC is adopted for model selection which penalises the number of parameters in the model.

Let $\mathcal{Y} = \{y_i \in \mathcal{R}^d : i = 1, ..., N\}$ be the data set we want to cluster; let $\mathcal{M}_k$ be the candidate parametric model which has k classes. Each class is modelled as a multi-variate Gaussian distribution. BIC of model $\mathcal{M}_k$ takes the form:

$$BIC(\mathcal{M}_k) = -\log(L(\mathcal{M}_k)) + \frac{1}{2}N_p(\mathcal{M}_k)\log N_s \tag{2.13}$$

where $L(\mathcal{M}_k)$ is the maximum likelihood estimate given the model $\mathcal{M}_k$, N_s is the number of samples and $N_p(\mathcal{M}_k)$ is the number of parameters in the model $\mathcal{M}_k$. The number of parameters for each class of $\mathcal{M}_k$ is $d + \frac{1}{2}d(d+1)$. Therefore,

$$N_p(\mathcal{M}_k) = k(d + \frac{1}{2}d(d+1)) \tag{2.14}$$

The BIC procedure is to choose the model with smallest $BIC(\mathcal{M}_k)$. The number which minimises the BIC value is denoted k_{BIC} in Section 2.5. The BIC criterion has been found by various authors to give reasonable results when selecting the number of components in a Gaussian mixture, e.g. [105]. In our case, images are RGB colour images and the Gaussian mixture models are 3D. Therefore $d = 3$ in Equation (2.14). Given the initial labelling, the parameters of the class-conditional Gaussians (Equation (2.10)) are estimated from the pixel data classified as belonging to that class (using MAP classification).

2.3.4 Summary

In summary, the MRF pixel labelling problem can be solved by using the $\alpha-$expansion algorithm iteratively. Each image labelling f is initialised by using

GMM, and the number of labels k is estimated by using BIC. α takes every label $k \in \{1, 2, ..., K\}$. For each label k, the max-flow algorithm is used to find the best expansion move and results in a new labelling $\hat{f}$. $\hat{f}$ is only accepted if it decreases the energy. Once the labelling changes, parameters are re-estimated. The algorithm is terminated when there is no smaller energy obtained. The algorithm is shown in Table 2.2.

Begin
1. Start with a labelling f with k labels chosen by BIC automatically;
2. For $\alpha = 1 : K$;
2.1. Set Flag=0;
2.2. Get a new labelling $\hat{f}$ by using Max-Flow;
2.3. If $E(\hat{f}) < E(f)$, set $f = \hat{f}$, Flag=1, and re-estimate the parameters;
2.4. If Flag=1 go to step 2.1;
3. End For;
4. Return f;
End

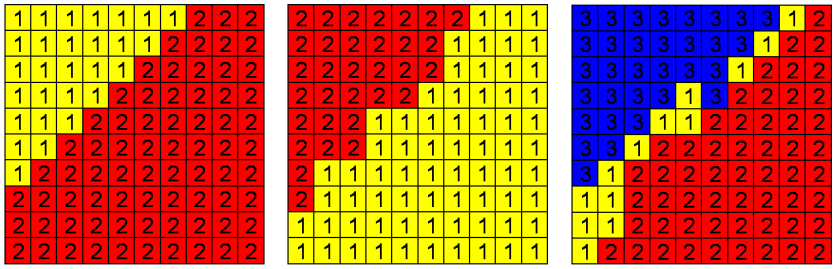
Table 2.2: Multi-label pixel labelling algorithm

2.4 Quantitative evaluation

Ground-truth labelling can be produced for each image by manual annotation. For visualisation, these labels were mapped to contrasting colours. The accuracy of an automated labelling was then evaluated by comparing it to the ground-truth. However, the label values were interchangeable. Furthermore, the labelling obtained will not generally have the same number of label values as the ground-truth. If two segmentations use N and M labels respectively then there are $R(M, N) = \frac{M!}{(M-N)!}(0 < N \leq M)$ possible mappings of the N labels in one segmentation to the M labels in the other segmentation. Note that $U = M - N$ labels will not find a match ($U \geq 0$). The measure of segmentation error which we use is the percentage of pixels labelled differently from the ground-truth in the mapping that minimises this proportion, i.e.,

$$err = \min_{\Pi} \frac{100}{m} \sum_{p=1}^{m} \delta(f_p^{(1)} - \Pi(f_p^{(2)})) \tag{2.15}$$

where $f_p^{(1)}$ and $f_p^{(2)}$ are labels assigned to the same pixel in the two segmentations, and Π denotes a permutation of the labels. Note that labels that do not find a match will contribute to this error. The percentage of pixels with unmatched labels will be denoted as $\mathcal{U}$.



(a) (b) (c)

Figure 2.11: (a) A ground-truth labelling of a 10×10 pixel image. (b-c) Two automatically produced labellings.

Figure 2.11 shows a simple example for the purpose of illustration. There are $R(2,2) = 2$ possible permutations between Figure 2.11(a) and 2.11(b): $\begin{pmatrix} 1 & \rightarrow & 1 \\ 2 & \rightarrow & 2 \end{pmatrix}$ and $\begin{pmatrix} 1 & \rightarrow & 2 \\ 2 & \rightarrow & 1 \end{pmatrix}$ where a $\rightarrow$ b means label value "a" in labelling 1 corresponds to label value "b" in labelling 2. Clearly, the latter produces a lower error. It is this error which is reported. In the same way, there are $R(3,2) = 6$ possible permutations between Figure 2.11(a) and Figure 2.11(c). The permutation that results in the lowest error is: $\begin{pmatrix} 1 & \rightarrow & 3 \\ 2 & \rightarrow & 2 \\ & & 1 \end{pmatrix}$. In this case, labels with value "1" in labelling 2 are unassigned and so contribute to the error. The accuracy of the labelling in Figure 2.11 is thus $err = 17\%$ of which $\mathcal{U} = 13\%$ is attributable to unmatched labels resulting from incorrect estimation of the number of labels, K.

2.5 Experiments and results

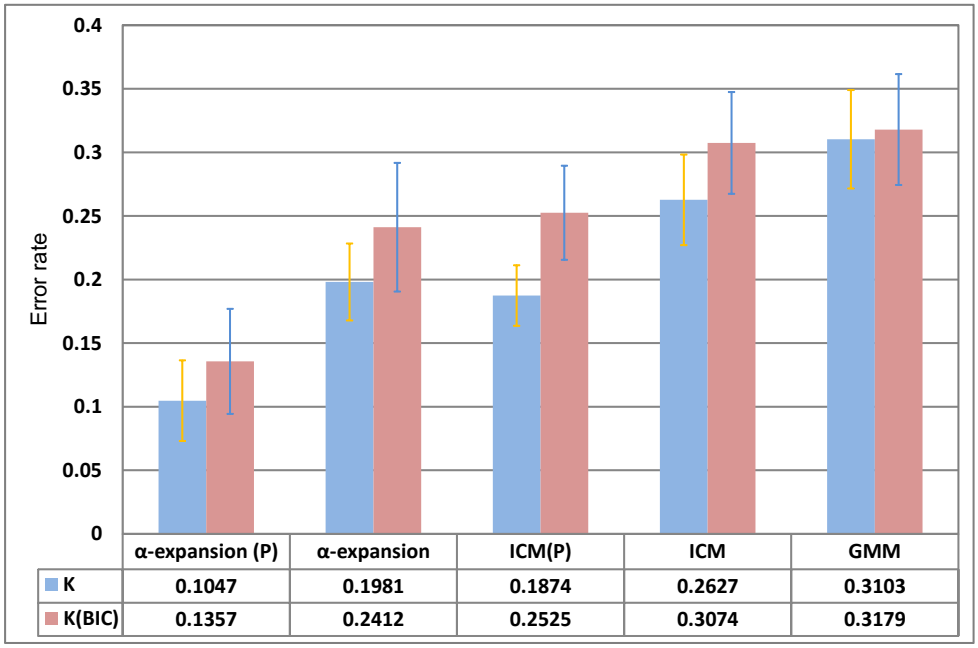


Figure 2.12: Mean segmentation error rate with known k and k_{BIC} on twenty example images. The error bars denote ± standard error of the mean segmentation error rate of the twenty images.

Five methods were evaluated on twenty example images from Liberty Art Fabric. Table 2.3 shows the twenty original example images and their ground truth images. The ground truth images were produced by manually assigning different colours to the pixels. The first method was Gaussian mixture model clustering using maximum likelihood EM parameter estimation. Each pixel was labelled with the most probable Gaussian component index conditioned on the pixel's RGB values. The second and third methods were MRF labellings found using ICM with and without parameter re-estimation, denoted ICM and ICM(P) respectively. The fourth and fifth methods were MRF labellings found using $\alpha-$expansion with and without parameter re-estimation, denoted α and α(P) respectively. The free parameter in the MRF models was set to λ=6 (see Equation (2.12)) for all images. We implemented $\alpha-$expansion with and without parameter estimation based on the two-label max-flow algorithm [116]. The test program was implemented in C and ran on an Intel Core 2 Quad Q6600 2.40GHz PC. Five methods were compared in two situations: (1) The number of labels (k) is assumed known. (There is information about the number of colour dyes used in some of the images of the Liberty Art Fabric database). (2) The number of labels is unknown (This is more difficult but more realistic situation), and estimated based on BIC values (k_{BIC}) obtained when EM was used to optimise

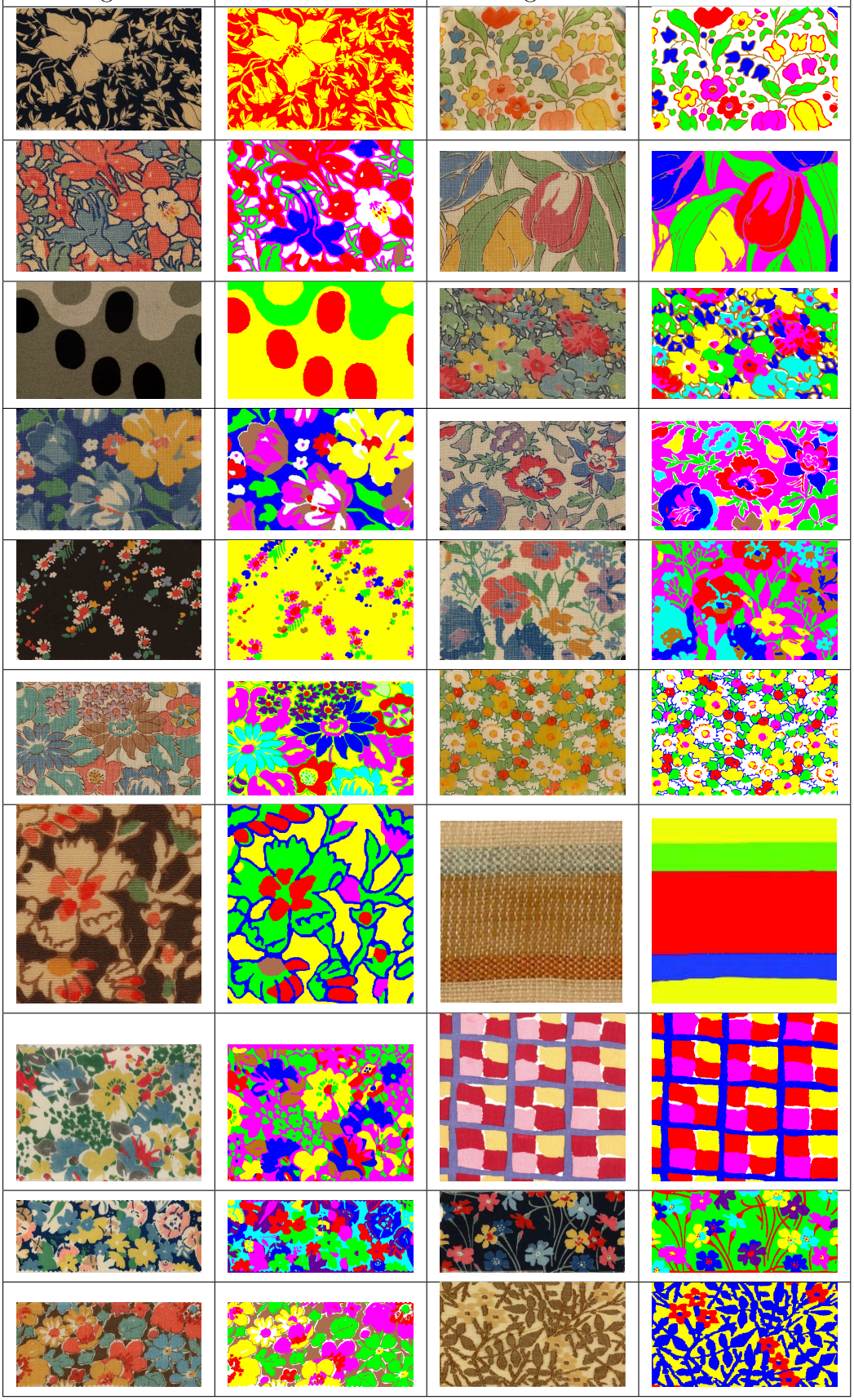
Original	Ground truth	Original	Ground truth
			

Table 2.3: Twenty original and ground truth images. (Image courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

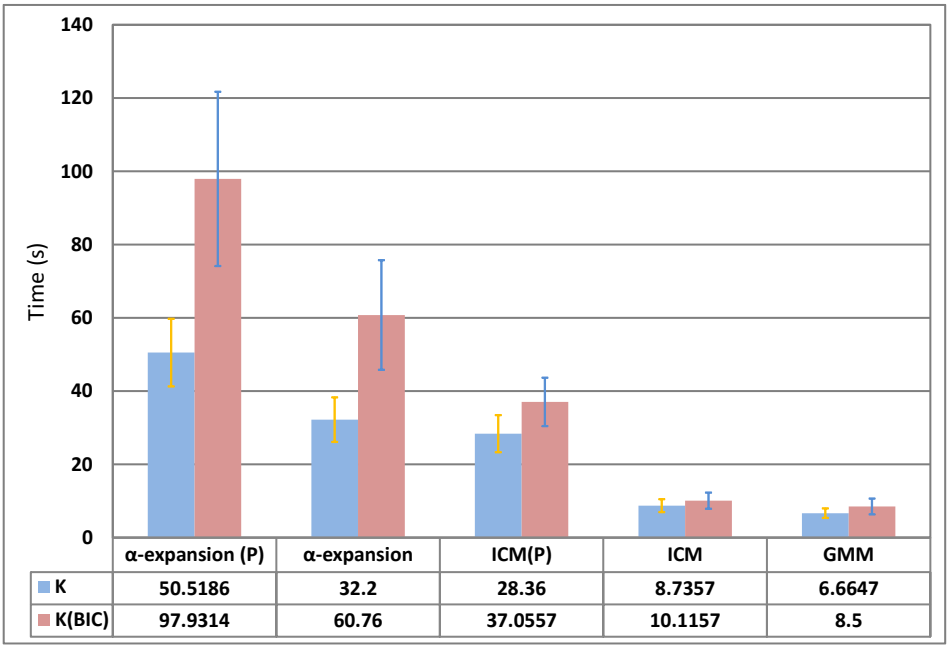


Figure 2.13: Mean times of five methods with known k and k_{BIC} on twenty example images. The error bars denote ± standard error of the mean times of the twenty images.

the parameters of the GMM. The number of colour dyes used is often limited by production costs in the type of textile production represented in the data set. Therefore, very large values of the number of labels can be ruled out *a priori*. Specifically, the range of number of labels from 1 to 10 was considered.

Figure 2.12 and Figure 2.13 report the mean segmentation error rate and the mean times with known k and k_{BIC} on the twenty example images shown in Table 2.3. The segmentation errors were calculated using the method in Section 2.4. In Figure 2.12, $\alpha-$expansion with parameter estimation gives the lowest error. ICM with parameter estimation is slightly better than $\alpha-$expansion without parameter re-estimation. This suggests that parameter re-estimation is important for these methods. GMM gives the highest error. In Figure 2.13 $\alpha-$expansion with parameter re-estimation is the slowest. ICM (P) is slower than ICM, α(P) is slower than α and GMM is the fastest. Note that the times reported for ICM and α - expansion do not include the time taken to initialise the labels using GMM-EM. For each algorithm, computational expense increases with k.

Tables 2.4 to Tables 2.10 show some typical images and their segmentations in detail, and Table 2.8 and Table 2.9 are the corresponding segmenation error and executive time of these images.

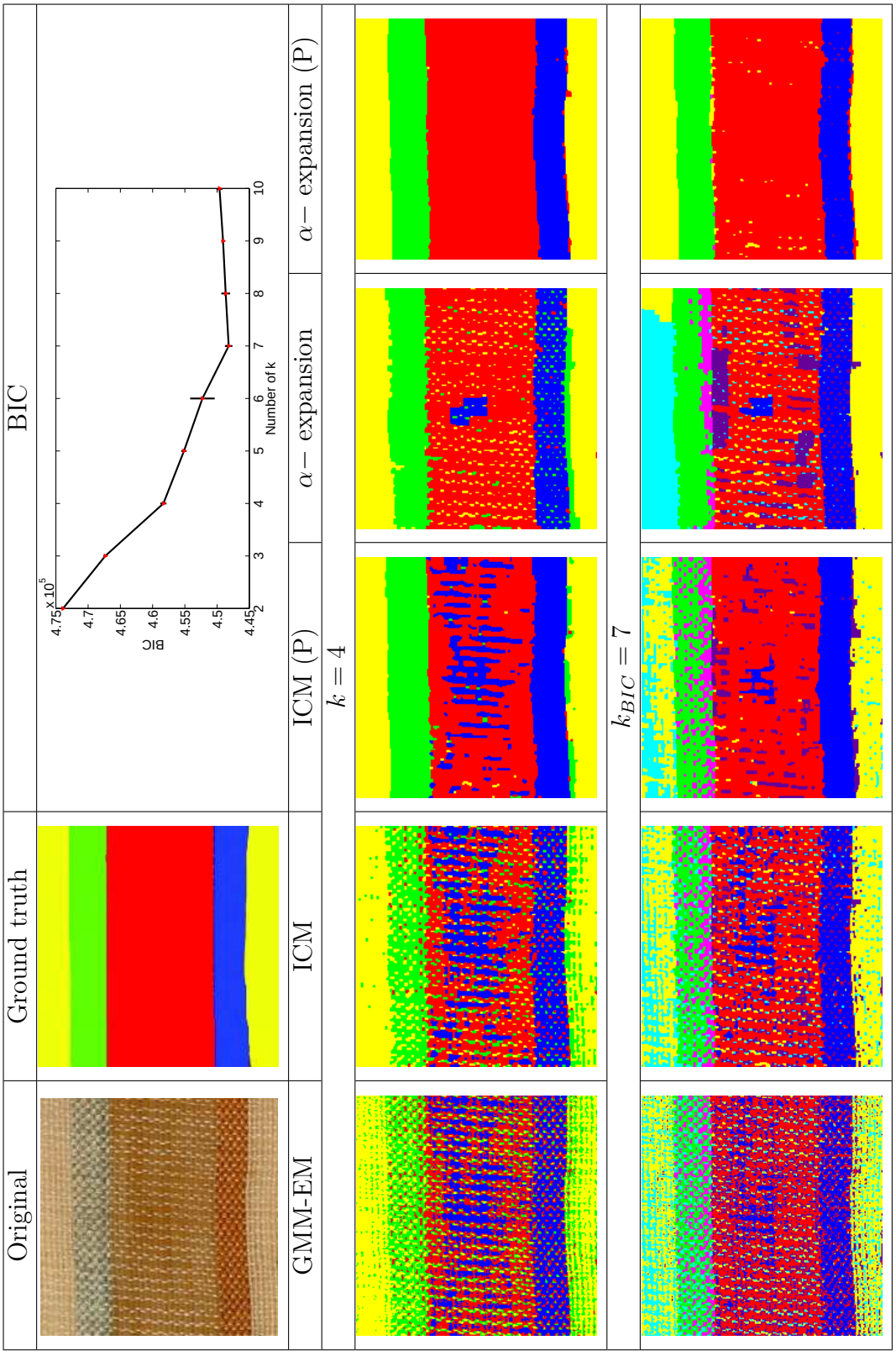


Table 2.4: (a) Segmentation results on a 200×200 textile image (b) BIC against number of k. (Image courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

There are four different colours of filling yarn (i.e., the yarn that runs horizontally) in the fabric in Table 2.4. However, the strong texture and uneven appearance resulted in BIC estimating k as 7. The graph plots the BIC value against k. Error bars denote $\pm$ a standard deviation (σ) estimated over 10 runs for each value of k. Segmentation results using $k = 4$ and $k_{BIC} = 7$ are shown. In both cases, $\alpha-$expansion with parameter re-estimation was clearly superior.

It is interesting to note that although k was usually overestimated by BIC, the effect of this on the segmentation error for $\alpha-$expansion with parameter re-estimation was not always large. For example, it can be seen that there is no large difference in error with different k in Table 2.4. $\alpha-$expansion can be understood as a competition between different labels. In every expansion, α takes one value from $\mathcal{L} = \{1, 2, ..., K\}$ and makes some of the pixels become α simultaneously. If the change is accepted, parameters will be re-estimated, a learning process. This helps to balance the percentage of different labels. The ground-truth of the original image in Table 2.4 contains four colours, while $k_{BIC} = 7$, but after $\alpha-$expansion with re-estimation only 0.4% of pixels are assigned labels 5, 6, or 7. Similar comments apply to the original images in Tables 2.6 and 2.7. Sometimes, larger k can help us to distinguish the ambiguous colours. For example, for the original image in Table 2.7, $k = 8$ was not enough to distinguish the brown and red colours due to the texture effects and dye degradation, but $k_{BIC} = 9$ could and resulted in a smaller error rate.

The design in Table 2.10 was not printed, but instead, filling yarns were inserted into the warp yarns to create the floral pattern. Here, when $k = 3$, α(P) is superior to the other methods, but when $k_{BIC} = 7$, although α(P) is better than others, it loses accuracy.

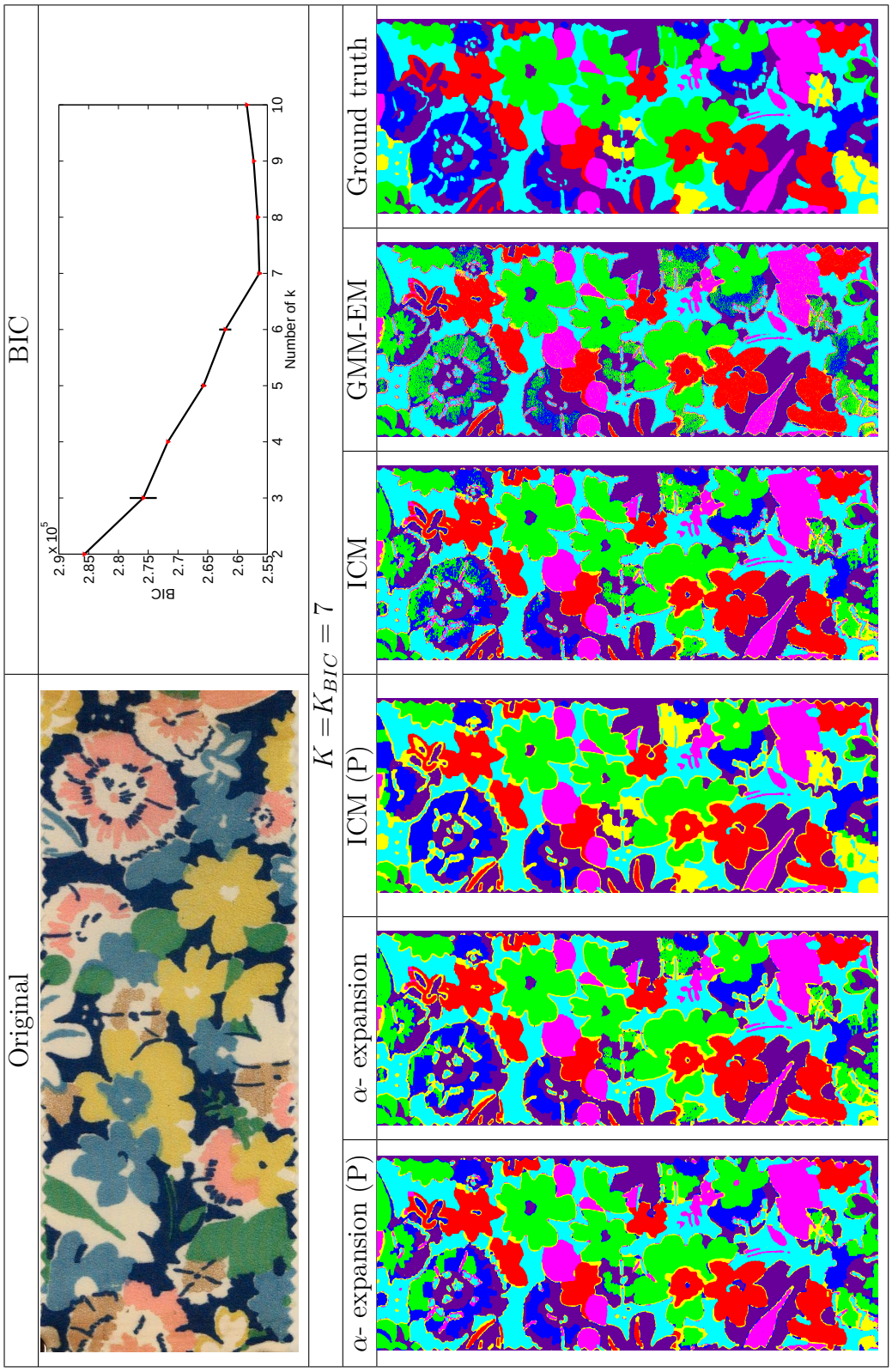


Table 2.5: (a) Segmentation results on a 1400×544 textile image (b) BIC against number of k. (Image courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

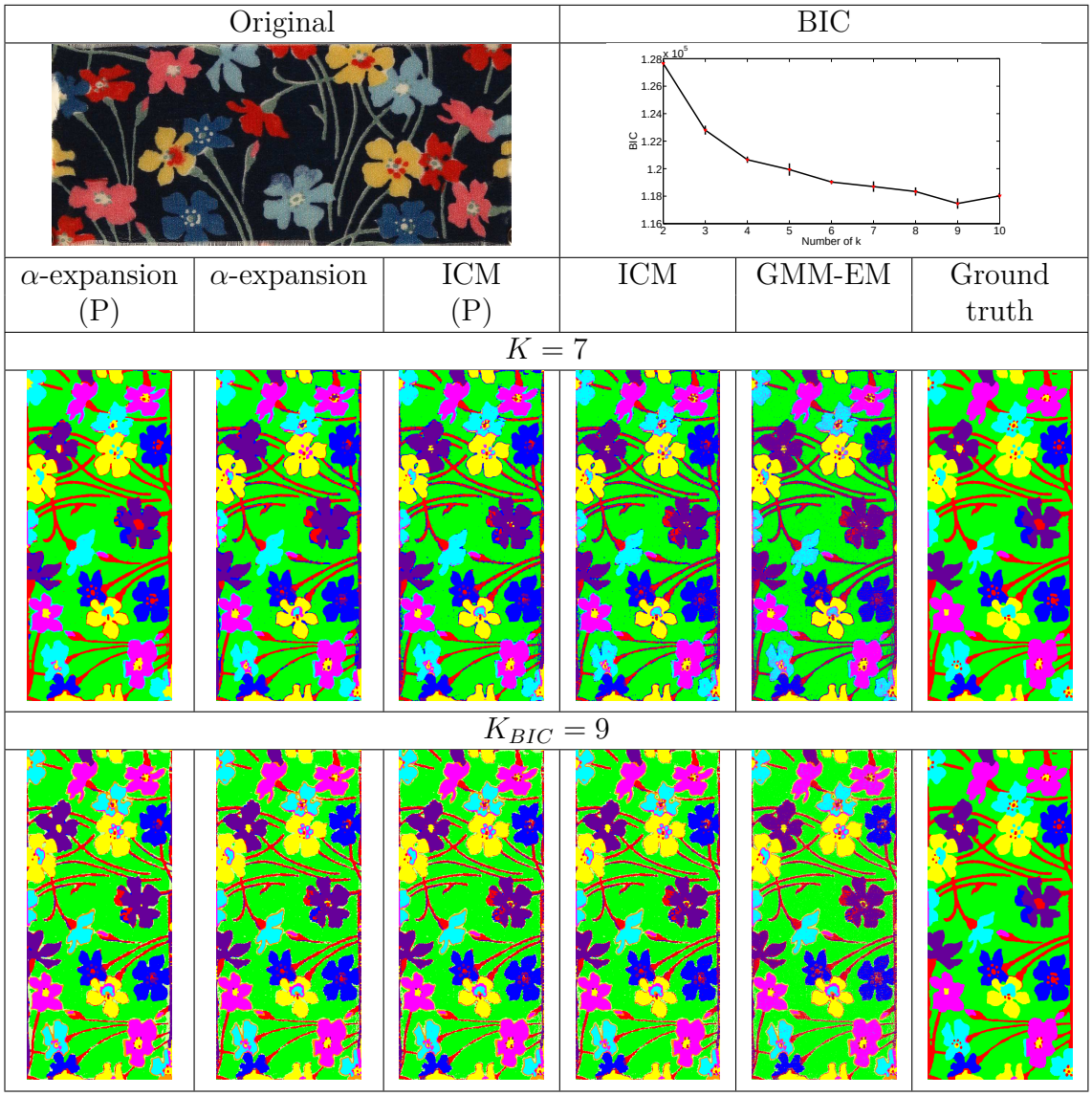


Table 2.6: Segmentation results on a 1268×556 textile image. (Image courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

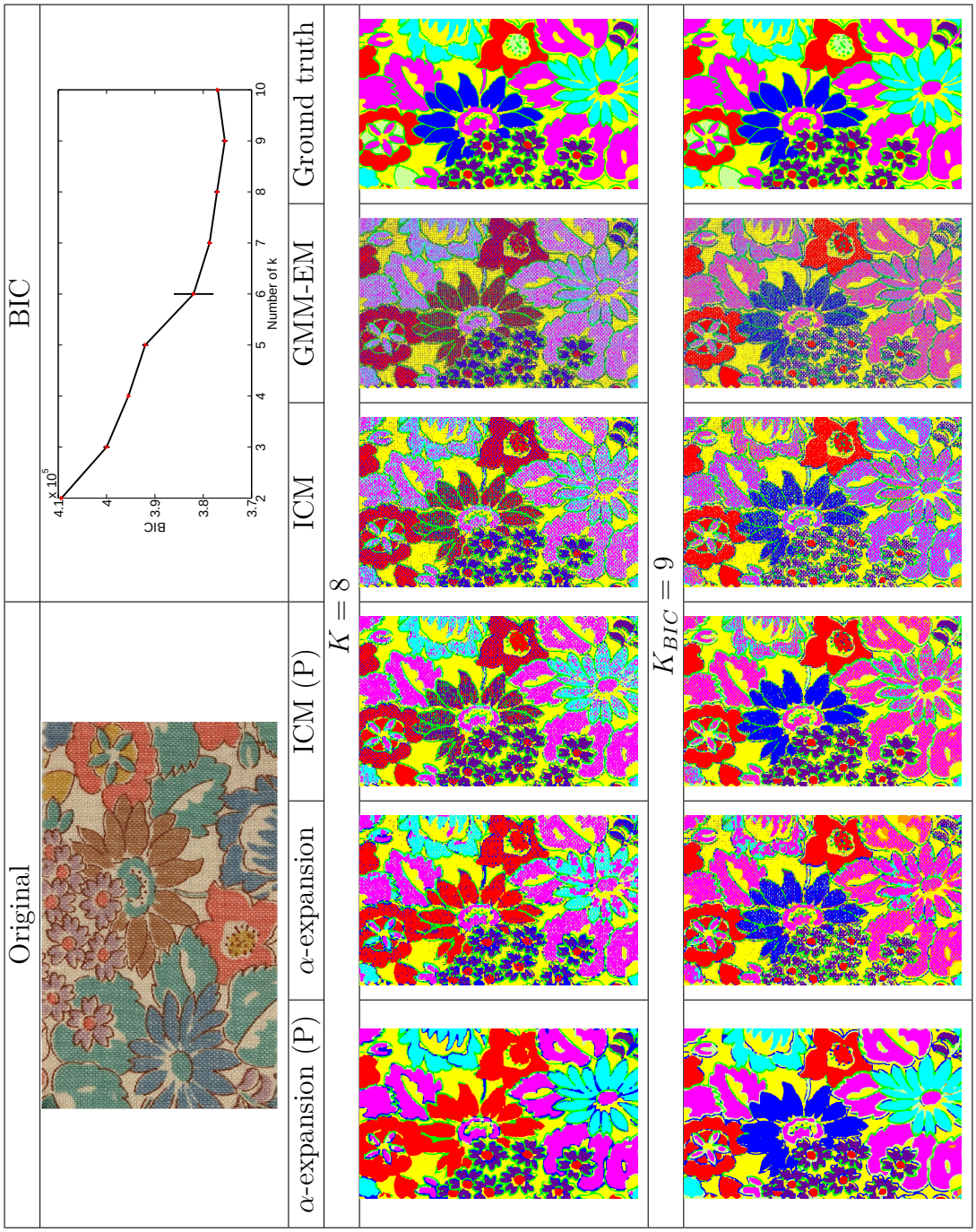
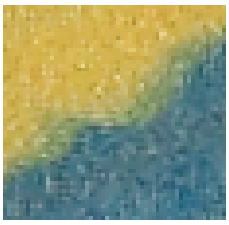
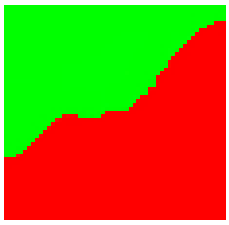


Table 2.7: Segmentation results on a 996×608 textile image. (Image courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

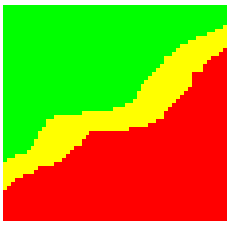
(a)



(b)



(c)



(d)

Figure 2.14: (a) Cropping small patch of original image (b) Original image patch (c) Ground truth (d) α-expansion result. (Image courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

Original Image in table #	Size	k	Error(%):err				
			GMM	ICM	ICM(P)	α	α(P)
2.4	200×200	4	39.07	30.85	13.56	11.68	3.30
2.5	1400×544	7	20.85	17.40	15.9	17.04	12.83
2.6	1268×556	7	17.05	14.27	12.23	12.9	6.63
2.7	996×608	8	48.99	42.89	31.56	34.18	24.99
2.10	1594×896	3	25.60	24.00	22.67	23.39	4.65

(a)

Original Image	Time(s)				
in table #	GMM	ICM	ICM(P)	α	α(P)
2.4	0.19	0.17	0.95	0.74	1.02
2.5	7.70	5.23	31.7	42.83	46.06
2.6	7.02	4.80	29.19	27.19	58.93
2.7	4.97	4.64	28.52	24.14	45.85
2.10	5.20	4.98	27.89	31.42	49.81

(b)

Table 2.8: (a) Error rates with known K (b) Computing times with known K

Original image in table #	k_{BIC}	Error(%):$\mathcal{E}[\mathcal{U}]$				
		GMM	ICM	ICM(P)	α	α(P)
2.4	7	44.97 [30.39]	35.00 [25.25]	19.99 [17.07]	32.75 [27.65]	5.52 [0.4]
2.5	7	20.85 [0.00]	17.40 [0.00]	15.9 [0.00]	17.04 [0.00]	12.83 [0.00]
2.6	9	18.31 [9.69]	16.12 [9.58]	15.05 [9.53]	15.39 [9.54]	8.37 [4.43]
2.7	9	17.79 [6.83]	40.75 [5.64]	30.62 [9.19]	34.66 [8.32]	19.01 [0.50]
2.10	7	45.15 [44.10]	42.99 [42.34]	45.27 [44.92]	46.76 [46.37]	37.53 [36.76]

(a)

Original image in table #	Time(s)				
	GMM	ICM	ICM(P)	α	α(P)
2.4	0.28	0.27	1.95	0.91	2.25
2.5	7.70	5.23	31.7	42.83	46.06
2.6	9.11	5.92	37.02	44.05	90.99
2.7	7.69	5.19	31.8	26.78	55.33
2.10	10.52	9.88	59.92	104.89	179.75

(b)

Table 2.9: (a) Error rates with k estimated automatically (b) Computing times with k estimated automatically

Tables 2.5 and 2.6 show less strongly-textured fabrics. Nevertheless, they are problematic. Figure 2.14 shows a magnified area where yellow and blue regions overlap and appear green. In order to avoid the gap between different colour regions, two dyes may be overlapped slightly, thus producing a third colour; causing difficulty in colour segmentation. Figure 2.14 graphically illustrates this problem. Figure 2.14(b) shows an annotated ground-truth. However, the α-expansion method separates the overlapping area into another colour region as shown in Figure 2.14(c).

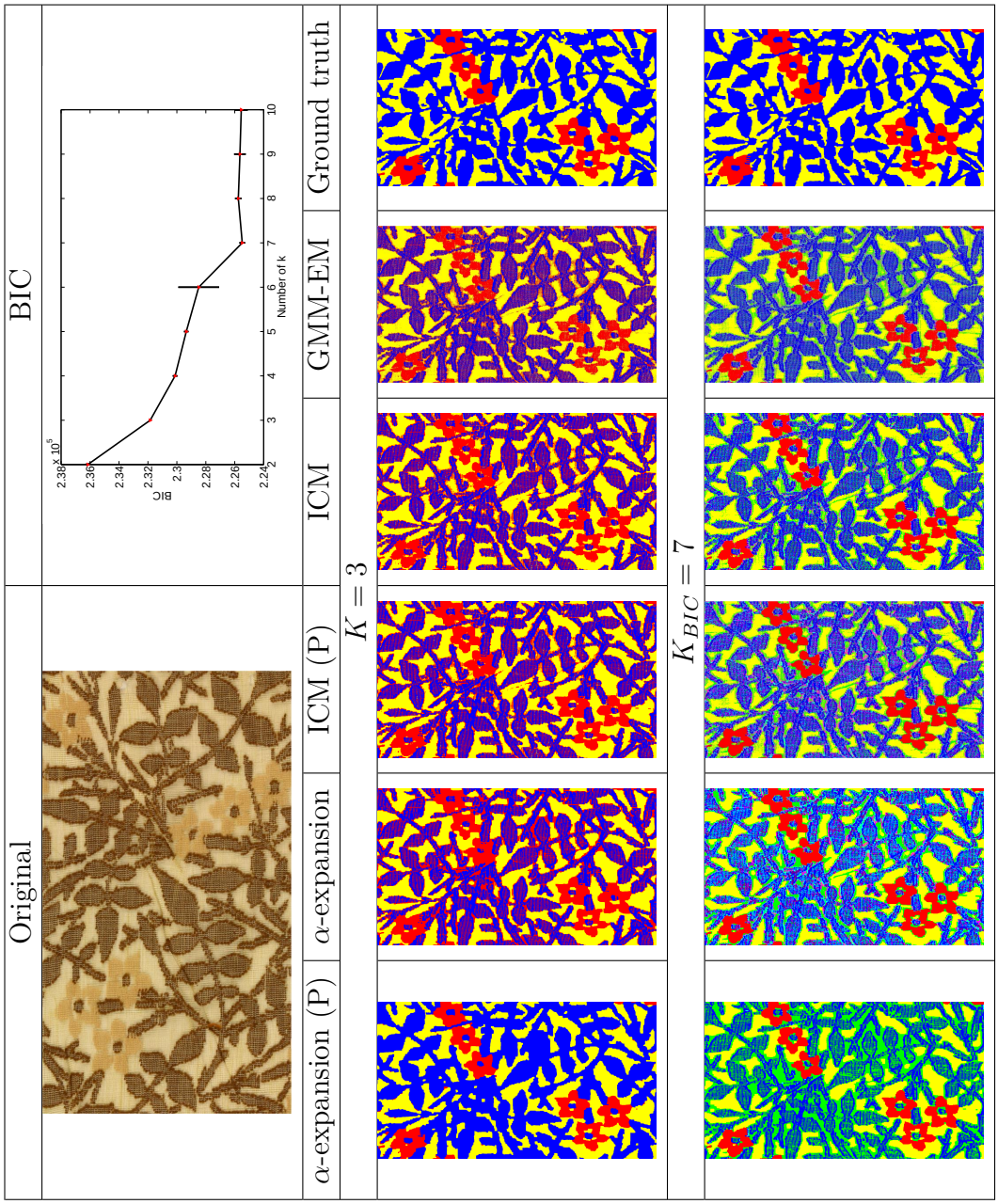


Table 2.10: Segmentation result on a 1594×896 textile image. (Image courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

2.6 Discussion and conclusions

In this chapter, we formulated the problem of extraction of printed design and woven patterns as MRF multi-pixel labelling in order to obtain smooth segmentation results. This model segmented the textile images into different homogeneous colour regions sharing similar RGB colour features. Compared with Gaussian mixture models, MRF pixel labelling can obtain smoother and more accurate segmentation results due to this method considering the spatial coherence in the first order neighbourhood system. However, it is slower than GMM. Finding optimal labels is equivalent to minimising the energy function of the MRF. A global method, α - expansion, and a local method, ICM, were compared to minimise the energy function. In both of the methods, parameter re-estimation can help to improve the accuracy. These algorithms were initialised based on GMM and BIC was used to select the number of distinct class labels. BIC based on GMM usually overestimated the number of class labels. However, this did not always result in $\alpha-$expansion with parameter re-estimation obtaining a less accurate result. Mismatch between number of dyes and number of actual colours are probably because: (1) some regions are undyed, (2) different colour dyes overlap and thus produce new colours, and (3) some of the textile designs are damaged or with labels on. For all the images tested, $\alpha-$expansion with parameter re-estimation was the most accurate and the most computationally expensive method. $\alpha-$expansion without parameter re-estimation gave similar performance to ICM with parameter re-estimation.

Choosing the number of class labels is an open problem. There is no agreed upon solution. Fortunately, the Liberty Art Fabrics database contains information about the number of colour dyes used in some of the textile images, so for image representation in chapter 3, the segmentation algorithm was run by obtaining the number from the database. The parameter λ of the energy function in Equation (2.7) specifies the penalty for assigning different labels to neighbouring pixels,

and decides the smoothness of segmentation. In this experiment, $\lambda = 6$ was used for all of the testing images, but for the experiment of textile classification in Chapter 4, the value was empirically set to 4 due to the different dataset.

The pixel labellings can be used for textile CAD (Computer-Aided Design) to redesign the textile image by assigning each label a colour dye. Pixel labellings also suggest ways of compactly representing image content in terms of colour and shape. Chapter 3 will present novel image representations based on MRF pixel labelling with α- expansion (parameter re-estimation), in order to classify textile designs in Chapter 4.

Chapter 3

Bags of shapes and region label graphs

3.1 Introduction

This chapter proposes bags of shapes and region label graphs that provide representations that could be used for various tasks such as image classification, categorisation, recognition, retrieval, visualisation, clustering, tagging, etc. The use of segmented regions for image recognition [4, 47, 51], and image categorisation [24, 9] has recently gained interest but strategies based on low-level patch descriptors such as SIFT [80] are often preferred [39], at least in part due to concerns about unreliable segmentation. In Chapter 2, reasonable extracted designs and patterns, namely, segmented region results can be achieved using Markov random field pixel-labelling. This method segments the images into groups of colour regions, where each group shares the same colour. Bag of visual words models are very popular in image representation [136, 98] due to their simplicity and good performance. The basic idea is to treat an image as a collection of independent regions or patches, describing each independent region or patch using

some visual descriptors, and using the distribution of sample regions and patches in the descriptor space as the characterisation of the image.

Motivated by bag of words models, we propose a bag of shapes model by considering MRF pixel labelling segmented regions as visual words. The regions are obtained from the labelling result by ignoring the colour label information and simply extracting the connected components. Every region is described by shape descriptor: Generic Fourier Descriptor (GFD), thus each image is represented as bag of shape descriptors.

To capture the spatial information, a graph is constructed based on region labels, which is presented in Section 3.4. Apart from partitioning an image into regions, MRF labelling identifies groups of regions that have the same label values. This representation uses the advantage of the pixel labelling, and constructs a graph by considering each group of regions as a vertex. Therefore, different images in the database may have different numbers of relationships between groups as edge weights. A series of unweighted graphs are obtained by removing edges in order of weight. Finally, an image is represented using its shape descriptors along with features derived from the chromatic numbers or domination numbers of the unweighted graphs and their complements.

The remainder of this chapter is structured as follows. Section 3.2 provides an overview of the related works on image representation and region shape descriptors. Section 3.3 describes the bags of shapes model. Section 3.4 provides another image representation method: region label graphs.

3.2 Literature review

3.2.1 Related work on bag of words model

A commonly used image representation approach is based on a global feature set

extracted from images, for example, images can be described by global distributions of colour, texture, or other properties, e.g. colour histograms [115, 114, 102], and Gabor filters [56] belong to this category. Unlike the global features, local features are based on subsets of images. (e.g. images are subdivided into (non) overlapping sub-blocks, or segmented into sub-regions by using some segmentation algorithms.) [122]. Recently, methods based on local image features have shown promise for texture and object recognition, and image classification [145, 15].

Recent literature contains many articles in which images are represented using geometry-free models: i.e. orderless bag of visual words models. Bag of visual words models were popularised in computer vision by Sivic and Zisserman [110] and Csurka *et al.* [29], and have proved to be effective for unsupervised image classification [39, 111, 103, 43], object classification [29, 131], video retrieval [110] and image annotation [95]. The success of orderless models for image classification can be understood with the help of an analogy to bag of words models for text document classification [86, 96]: a document is a bag of words. The most relevant words are found to describe a document class by assigning weights to the words, and then documents are effectively matched through matching functions.

Constructing the bags of visual words for images contains the following steps:

- Automatically detect interesting regions or points (local patches) which contain rich local information of an image. Many detectors, e.g. Hessian-affine and Harris-affine, find comparable types of regions, basically regions containing corners or changes in lighting [91]. By contrast, some detectors, typically, the Maximally Stable Extremal Region (MSER) detector, detect regions based on their stability. .

- Compute local descriptors of the regions or points (e.g: SIFT, SURF [90]), and the regions are then described as multidimensional vectors.

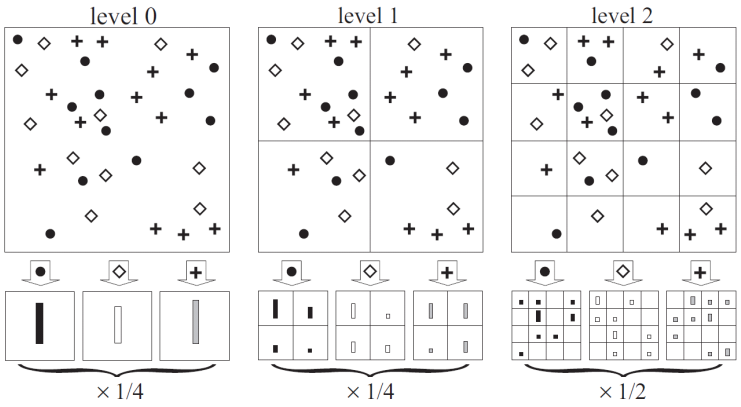


Figure 3.1: This is a simple example of constructing multi-level spatial pyramid from Lazebnik *et al.* [75]. Images are subdivided at three different levels of resolution. For each level and each cell, features that fall in each spatial bin are counted, and each spatial histogram is assigned a weight according to the weight equation in Lazebnik *et al.* [75].

- Obtain the visual vocabulary by descriptor quantisation into codewords using clustering algorithm. Each cluster corresponds to a visual word. Many studies about the clustering algorithms were published [97, 100, 74] and it is shown that it has significant impact on the results [97, 100]. The choice of clustering algorithm is important since the calculation of codewords is very much dependent on it. However, it is also an issue that some descriptors are high-dimensional, e.g, SIFT has 128 dimensions.

- Find the occurrences of each specific word in the vocabulary to build the bag of words for each images.

The bag of words model has the advantage of simplicity, but it discards spatial information. Since the bag of words model depends on a count of the visual word occurrences in the images, the relationships between words are ignored. Such relationships seem important. Two different objects can share similar words, but the layout of the words is unique to each object.

Much work has been done on adding geometric information to the bag of words model. Zheng and Gao [146] propose a visual phrase-based approach which

groups visual words into visual phrases by grouping pairs of adjacent local image patches. For speed, they only consider frequent pairs whose frequency in the image collection is above a certain threshold rather than testing all possible pairs of words. Yuan *et al.* [139] generate higher-level visual phrases with arbitrary numbers of words per phrase, using a k nearest-neighbourhood algorithm to group words, and then post-processing to extract more relevant and general patterns. Tirilly *et al.* [120] present visual sentences which read visual words in a certain order, as in the case of text to consider simple spatial relations between words, and then use language models to exploit these relations. To encode spatial information, Yang *et al.* [136] divide an image into equal-size rectangular regions (e.g. 4×4 means cutting an image into 16 regions in 4 rows and 4 columns), compute visual word features from each region, and concatenate the features of these regions into a single feature vector. However, dividing images into sub-regions may lead to mismatching. For example, suppose an image class is defined by the presence of an object, say bus, which may appear anywhere in the image. If the bus appears in training images in different regions from that in testing images, that may cause feature mismatching. Therefore, coarse partitions were preferred to be considered rather than fine partitions. Lazebnik *et al.* [75] construct multi-level pyramids which partition an image into increasingly fine grids and compute histograms of patches found inside each grid cell (see Figure 3.1). For each level of resolution and each cell, weighted histograms of codewords are calculated. The feature of one image is the concatenation of all of the weighted histograms of all of the grid cells at all of the level resolutions. The resulting spatial pyramid showed improvements over orderless bag-of-words representations. Different from [75], Battiato *et al.* [7] partition an image using three different schemes: horizontal, vertical and regular grid (see Figure 3.2). However, these representations have their downside in cost because dividing an image into sub-regions increases the feature dimension and makes the feature computation relatively expensive.

A rigid partition of an image into fixed-size blocks usually breaks an object into

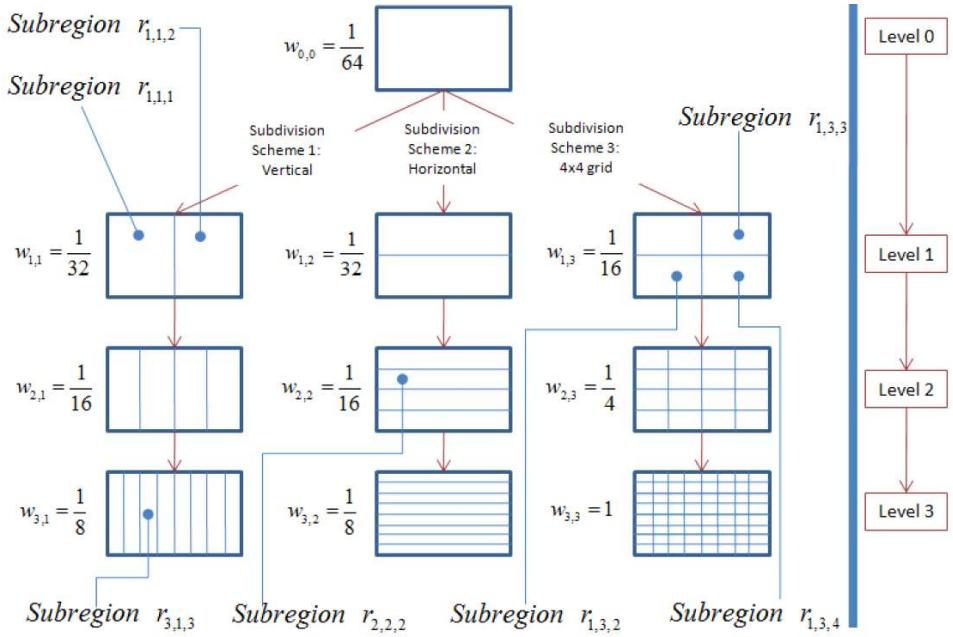


Figure 3.2: An example of a hierarchy of four level sub-regions from [7]. $r_{l,s,i}$ denotes the $i-$th sub-region at $l-$th level in subdivision scheme s. The weight $w_{l,s}$ is assigned to each sub-region as: $\frac{N_{l,s}}{N_{L,S}}$, where $N_{l,s}$ is the number of sub-regions of level l and in scheme s, and N_{max} is the maximum number of sub-regions among all of the levels and schemes.

several blocks or puts different objects into a single block. The visual information of the objects may be affected by the rigid partition. Therefore, in some cases, the visual words of bag of words can be regions segmented by region segmentation algorithms [22, 58]. Segmentation algorithms inevitably make mistakes. However, segmenting an image makes us access the level of objects, which contains important information for image classification and object recognition. Besides, the segmented regions are close to human perception and can be used as the basic blocks for feature calculation and similarity measurement. Carson *et al.* [22] presents "Blobworld", a framework based on representing images as groups of regions. Each region is described by using a combination feature of colour, texture and position. Image retrieval is then obtained by querying the properties of these regions. Jing *et al.* [58] represent images with segmented regions obtained from the JSEG algorithm. The regions are described using two properties: the visual features extracted from the regions and the weight of importance. Usually, the number of regions in this representation is not that big, and the regions some-

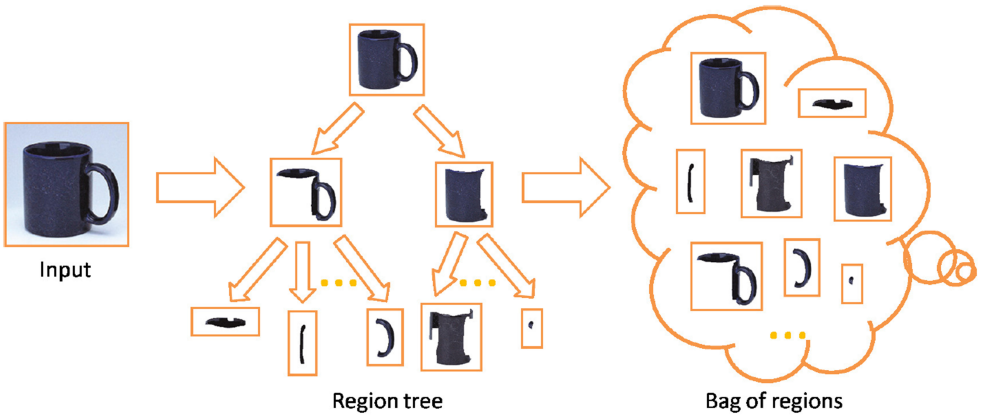


Figure 3.3: An example of using bag of segmented regions from Gu *et al.* [47]. A region tree of an input is generated by using segmentation algorithm (Arbelaez *et al.*) first, and all of the segmented regions are then collected to form the bag of regions with ignoring the tree structure.

times correspond to objects or parts of objects, e.g. horse or grass in the natural images. Gu *et al.* [47] construct a region tree for each image using the hierarchical segmentation of [4], and represent the image by using a bag of regions model in which the regions are the nodes of the region tree. Therefore, the range of the regions is from super pixel to the entire image. However, they ignore structure information of the region tree, and simply combine those regions together (see Figure 3.3). The region cues are the combination of contour shape [81], edge shape, colour and texture. This representation was applied in the application of recognition by Gu *et al.* [47].

In the case of bag of segmented regions models, an image is segmented into many regions which are in relationships with each other, but such spatial information is not often exploited. Gokalp and Aksoy [43] propose a bag of region pairs representation. In their work, images are partitioned into regions using one-class classification and a patch-based clustering algorithm at the first step. Secondly, the segmented regions are clustered to obtain a codebook. Images are then represented by a bag of region pairs representation in which regions satisfying a particular relationship are grouped together. The particular relationship they use is the vertical relationship of above-below. An overlapping criteria is used to de-

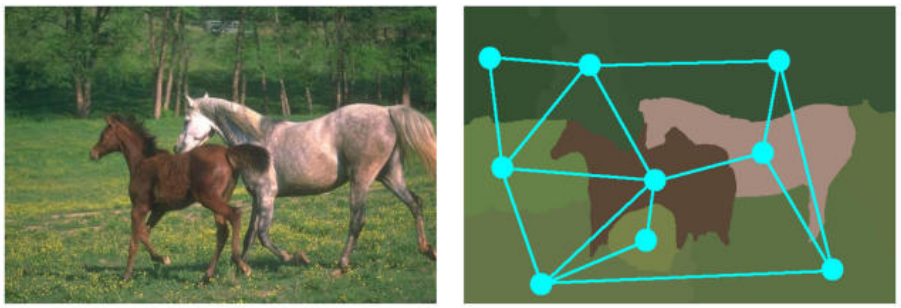


Figure 3.4: An example of a natural image (left) and its segmentation graph representation (right) from Harchaoui and Bach [51].

termine a list of region pairs that satisfy the "above-below" relationship. Many other different methods have been proposed to model the spatial relationships between regions ([3, 12, 70, 16]). de Mauro *et al.* [30] propose a graph-based image representation. First, the image is segmented into regions using a region growing algorithm, and each region is described by a real-valued vector of features. Then an intermediate structured representation is built as a Region Adjacency Graph (RAG) by associating the segmented regions to nodes in a graph, and encoding the spatial positions of the regions as the graph edges. General adjacency relationships of RAG can be horizontal, vertical, diagonal, included-in and centred-into. Based on RAG, de Mauro *et al.* [30] derive a directed ordered acyclic graph (DOAG) by defining a direction and an ordering for the edges in the graph. The DOAG representation can encode both the features of each region and relationships among the regions. Harchaoui and Bach [51] represent an image by a segmentation graph, an undirected labelled planar graph, in which each vertex is one segmented region using morphological segmentation [87] with edges joining neighbouring regions. Although Harchaoui and Bach [51] construct segmentation graph for different use, to compute the graph kernels for image classification, this method inspires us to use graph representation to extract spatial information from images.

Since reasonable segmentation results can be obtained using MRF pixel labelling method described in Chapter 2, we will propose an image representation using

a bag of MRF segmented regions. Spatial information between local regions or patches are shown to be important in the related works. Therefore, we attempt to encode the spatial information between the MRF segmented regions. To encode spatial information, graphs are constructed. However it is not realistic to construct graphs considering each segmented region as a graph node since the number of MRF segmented regions for each image is large. (about 500 on average for each image). Taking advantage of MRF resulting colour labels, each group of regions sharing the same colour label is considered as a graph node, and relationships are encoded into the edges of pairs of groups' regions.

Many features can be used to describe the segmented regions, e.g. colour and texture features. Shape is a key attribute of segmented image regions and characterising shape critically depends on reliable segmentation. Its efficient representation plays an important role in many applications. The shape descriptors discussed here are for segmented regions rather than contour, since contour-based shape descriptors may not be suitable for complex shapes (e.g. shapes consist of several disjoint regions) [63]. Section 3.2.2 will review the region shape descriptors.

3.2.2 Related work on region shape descriptors

Generally, a shape descriptor converts a 2-D shape into a fixed multidimensional vector such that similar shapes have similar shape descriptors. A shape descriptor should not be affected by rotation, scaling and translation. There is no need to reconstruct the shape from the shape descriptor but the shape descriptor should be discriminative enough to distinguish different shapes. Six requirements for good shape descriptors in the MPEG-7 standard [85] are: good retrieval accuracy, compact features, general application, low computational complexity, robust retrieval performance, and coarse-to-fine hierarchical representation [71].

Common region-based methods use moment descriptors to describe shape. The pioneering work in invariant moments was presented by Hu [54]. He employed the theory of algebraic invariants to derive seven moments, namely, Hu moments, that are invariant under translation, scaling and rotation. Flusser [40] presented a method for constructing invariants of arbitrary order by using complex moments and the problem of the independence and completeness of rotation moment invariants was addressed. These moment invariants based on geometrical moments have been used in many applications [25, 134, 128].

However, a disadvantage of these moments is that the invariants are not derived based on an orthogonal basis, which leads to a high degree of information redundancy. To overcome this problem, Teague [118] uses continuous orthogonal polynomials, such as Legendre and Zernike polynomials, to derive Legendre and Zernike moments [119]. Zernike moments are demonstrated to outperform other geometrical and Legendre moments [119], and to be highly effective in terms of image representation [63, 141]. MPEG-7 has selected Angular Radial Transform (ART) descriptors [13] as region-based shape descriptor. The ART descriptors are also based on moments, but they are based on cosine functions rather than on Zernike polynomials.

Recently, much research has been done on discrete orthogonal moments [147]. Mukundam *et al.* [94] proposed a set of orthogonal moments based on the discrete Tchebichef polynomials, and claimed that Tchebichef moments are superior to the continuous moments because the basis set is orthogonal in the discrete domain of the image coordinate space and since the implementation of the moments does not use any numerical approximation. Mukundam [93] also proposed radial Tchebichef moments to derive rotation invariants from Tchebichef moments. Other orthogonal-discrete moments proposed are: the Krawtchouk moments [137], the Racah moments based on discrete Racah polynomials [147], and the dual Hahn moments [148].

The Fourier transform is a powerful technique for shape analysis, but most of the Fourier shape descriptors are based on a 1-D Fourier transform and are mainly used for shape contours. However, Zhang and Lu [143] use the 2-D Fourier transform to derive a set of invariant descriptors for region shapes. In this method, a shape image in polar space is converted into Cartesian space and 2-D Fourier transform is then applied to obtain the shape descriptors. Li *et al.* [79] propose a modified version of the GFD that operates on edge information for natural image retrieval. El-Ghazal [38] point out that GFD is one of the most promising descriptors because it is based on the well-known Fourier theory, which simplifies the implementation and the interpretation of the results. Besides, GFDs are well suited for image retrieval or classification due to their compactness and computational efficiency.

3.3 Bags of shapes

This section proposes a new image representation method: the bags of shapes model, which is motivated by the popular bag of visual words model. In this model, instead of obtaining regions by using a region detection algorithm (e.g. [91]), regions are obtained by using the MRF pixel labelling method described in Chapter 2. Figure 3.5 shows an example of a pixel labelling and bag of shapes model. In this case, there are seven colour labels in the original image (left). If the colour labels are ignored, the image is represented as a single bag of regions with each region represented by a feature vector characterising its shape. Here, each region is described by a region shape descriptor: a generic Fourier descriptor (GFD). Thus, an image is represented as a *bag of shapes*. In the standard bag of words model, the codewords are calculated by quantising the features of the sample words using some clustering methods, and images are represented by histograms of the codewords. Feature quantisation can reduce the feature dimensionality to get a compact representation. However, it weakens the discriminative

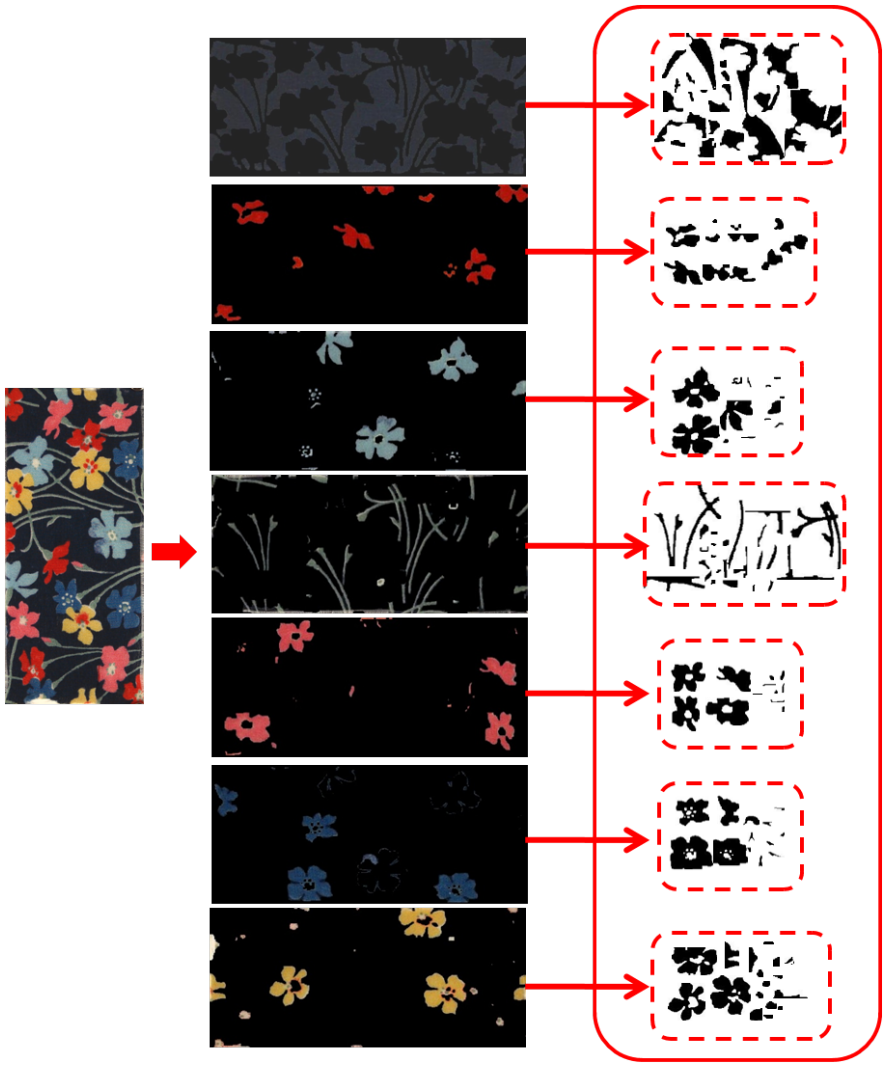


Figure 3.5: Left: the original image. (Image courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted) Middle: MRF multi pixel labelling result. The original image is segmented into seven homogeneous groups according to similar colours. Right: bag of shapes. Each colour group contains multiple regions and can be extracted to corresponding binary connected components (region shapes). All of these binary connected components form the bag of shapes model by ignoring the colour label information and mixing all of the region shapes together.

power of the feature descriptors. In our case, the features of an image are represented in an $S \times T$ matrix where S is the number of shapes and T is the length of the shape feature vector. (A suitable classification method for this kind of representation will be introduced in Chapter 4). Different images have different numbers of regions according to the segmentation. The importance of a region can be related to its area [77]. Regions with very small area are ignored here and an area threshold is set up to choose the regions (connected components). The regions are ordered in descending order according to their area, and a fixed number of regions is chosen to represent each image. This number is chosen empirically (Chapter 4).

3.3.1 Region shape descriptors

In Section 3.2, a lot of region shape descriptors were reviewed. The Generic Fourier descriptor satisfies the six requirements of MPEG-7 mentioned in Section 3.2.2. Besides, two more characteristics which distinguish it from other shape descriptors in the literature are: stability (stable performance in different applications) and clarity (clear physical meaning) [142]. GFD is proved very suitable for image retrieval and classification. Therefore, the author chose GFD as the shape descriptor to describe the region shapes in bags of shapes model. This section will introduce polar Fourier transform and the derived generic Fourier descriptors.

3.3.1.1 Polar Fourier Descriptor

The discrete 2-D Fourier Transform (FT) is a very useful transform in image processing. Features captured by 2-D FT are not rotation invariant. Rotation invariance is important as similar shapes may be oriented differently. Figures 3.6(a) and 3.6(c) are the same pattern, but rotated by 90 degrees. Their Fourier coefficients are different which makes it difficult to recognise that these two

patterns are the same.

To overcome this problem, Zhang and Lu [143] represent a shape image in the polar space, and convert it into the Cartesian space to get a norm 2-D rectangular image. Next, they apply a 2-D Fourier transform to the rectangular image. The polar Fourier transform in Cartesian space has the form:

$$F(\rho,\phi)=\sum_{r}\sum_{i}f(r,\theta_i)\exp[-j2\pi(\frac{r}{R}\rho+\frac{2\pi i}{T}\phi)], \tag{3.1}$$

where $0 \leq r < R$ and $\theta_i = \frac{2\pi i}{T}(0 \leq i < T)$. ρ and θ are the radial frequencies and angular frequencies respectively. The determination of ρ and θ are physically achievable since shape features are obtained by few low frequencies. Figure 3.6(e) and (g) show the polar image of the two patterns (Figure 3.6(a) and (c)) and Figure 3.6(f) and (h) are their corresponding Fourier spectra. It is observed that rotation of a pattern in Cartesian space (3.6(a) and (c)) results in the polar image being shifted, but the shift of polar image does not change the spectra of Fourier transform, which can be seen in Figure 3.6(f) and (h). Figure 3.7 shows an example of one of the region shapes extracted from the segmentation and its corresponding polar Fourier spectra. The polar images in the Cartesian space are produced using rectangular to transform code [82].

3.3.1.2 Derivation of Generic Fourier Descriptor

Given a shape image $I = f(x, y)$ where $0 \leq x < M, 0 \leq y < N$, the shape is represented in polar space, so $I_p = f(r, \theta)$ where $0 < r < R, 0 < \theta < 2\pi$, and R is the maximum of the radius. Since the origin of the polar space is the centroid of the shape image, the shape is translation invariant. The centroid (x_c, y_c) is:

$$x_c=\frac{1}{M}\sum_{x=0}^{M-1}x, y_c=\frac{1}{N}\sum_{y=0}^{N-1}y \tag{3.2}$$

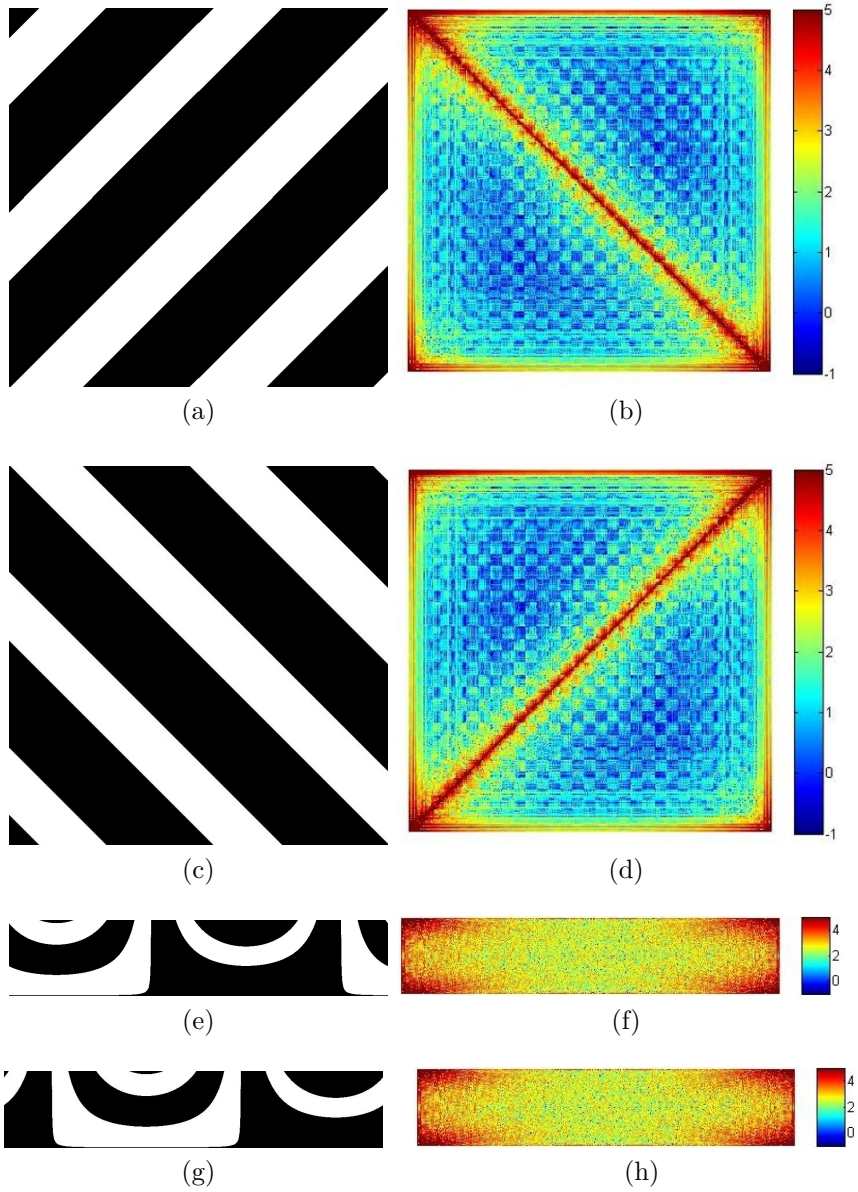


Figure 3.6: (a) A pattern (b)2-D Fourier spectra (c) Rotation 90°of (a) (d) 2-D Fourier spectra (e) Polar image of (a) in Cartesian space (f) 2-D Fourier spectra of (e) (g)Polar image of (c) in Cartesian space (h) 2-D Fourier spectra of (g)

Figure 3.7: Polar to Cartesian transformation of a region

and

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2}, \theta = arctan\frac{y - y_c}{x - x_c} \tag{3.3}$$

Rotation invariance is achieved by using only the magnitude of the coefficients and ignoring the phase information in the coefficients. To obtain scale invariance, the first magnitude value is normalised by the area of the circle in which the polar image resides or the mass of the shape, and all of the other magnitude values are normalised by the magnitude of the first coefficient. The shape descriptor is derived as Equation (3.4):

$$\mathbf{d} = \{\frac{|F(0,0)|}{area}, \frac{|F(0,1)|}{|F(0,0)|}, ..., \frac{|F(m-1,n-1)|}{|F(0,0)|}\} \tag{3.4}$$

where m is the number of radial frequencies and n is the maximum number of angular frequencies. For efficient shape description, only a small number of descriptors are selected for shape representation, and the selected descriptors form a feature vector which is used to index the shape. To calculate shape description, $m = 4$ and $n = 12$ are chosen to give a 48-dimensional feature vector, since the choice of m, n is shown to be efficient and effective to describe a shape in shape retrieval [143].

3.3.2 Discussion

In summary, the bags of shapes model for textile images is a collection of all of the segmented regions (connected components of every colour label class) from MRF pixel labelling. The regions are described by using 48 dimensional region shape descriptors, GFD. In standard bag of words models, codewords are computed by feature quantisation and images are represented as the histogram of codewords. The disadvantage of feature quantisation is that it weakens the discriminative power of feature descriptors. Thus, without codeword calculation and representing images by histogram of codewords, images are expressed by $s \times 48$ feature

matrices, where s is the number of region shapes contained in the image. The value of s differs in different images. A suitable classification approach for this kind of representation will be used for textile design classification. This will be further discussed in Chapter 4. Obviously, this representation ignores colour label information, e.g. at the bottom of Figure 3.5, region shapes in each dash line box share the same label. This is important information since shapes that share the same label are usually similar patterns and have similar properties which may be helpful to classify textile designs for example. Another disadvantage of the common bag of words model is the lack of the spatial information between single words. When visual words are segmented regions, many graph-based methods (e.g. [30]) are proposed to capture the spatial relationship between the segmented regions. These methods consider each segmented region as a node in the graph, and encode the relationship on the edges between different segmented regions. Generally speaking, these methods can deal with the segmentation with small number of regions due to the complexity. However, in our case, the number of segmented regions obtained from MRF pixel labelling is often more than 500. The graph-based method which considers each segmented region as a node may not be feasible because the number of the nodes may affect the computation complexity. Section 3.4 will propose a novel graph-based method, the region label graphs, which considers a group of regions sharing the same label as a node of the graph.

3.4 Region label graphs

This section proposes a novel image representation, using the potential colour label information which the bags of shapes ignore. As discussed in section 3.3.2, it may not be realistic to construct a graph with a node for every segmented region since the number of regions segmented from MRF pixel labelling is large. However, MRF pixel labelling identifies groups of regions that have the same label

values. The approach proposed here seeks to represent relationships between these groups of regions that might be useful for tasks such as classification. This leads to a weighted graph in which vertices are associated with region groups and edge weights indicate relationships between the groups. The edge weights can encode information on spatial adjacency or alternatively the dissimilarity of the regions in the groups. Such a representation differs from the traditional Region Adjacency Graphs (RAG) [30] since the vertices correspond to groups of regions rather than individual regions. Consequently, these graphs tend to have far fewer vertices. A method for obtaining feature vectors of fixed dimensionality from these graphs is proposed so that standard machine learning methods can be used. This involves generating series of unweighted graphs from the weighted graphs via edge removal. Feature vectors can then be based on graph-theoretic measures such as chromatic number and domination number.

3.4.1 Graph construction

To capture the spatial information in the image, graphs based on region labels are constructed. Figure 3.8 shows an example of how the region label graph is constructed. Firstly, an original image is segmented into groups of regions by assigning each pixel a label through the MRF pixel labelling procedure. The label image which is produced by using contrasting colours is shown in the centre. Each colour represents a group. Given such a labelling, the image is represented as a *bag of shapes* as described in Section 3.3. As discussed previously, this model ignores relationships between the groups of regions. In order to retain information about these relationships, we construct undirected weighted graphs as shown at the bottom of Figure 3.8. Each group of regions is considered as a node of a graph, and represented as a bag of shapes respectively. Each shape in each group is described by using a 48 dimensional GFD. Edges in the graph encode relationships between the groups. For example, edges can encode the

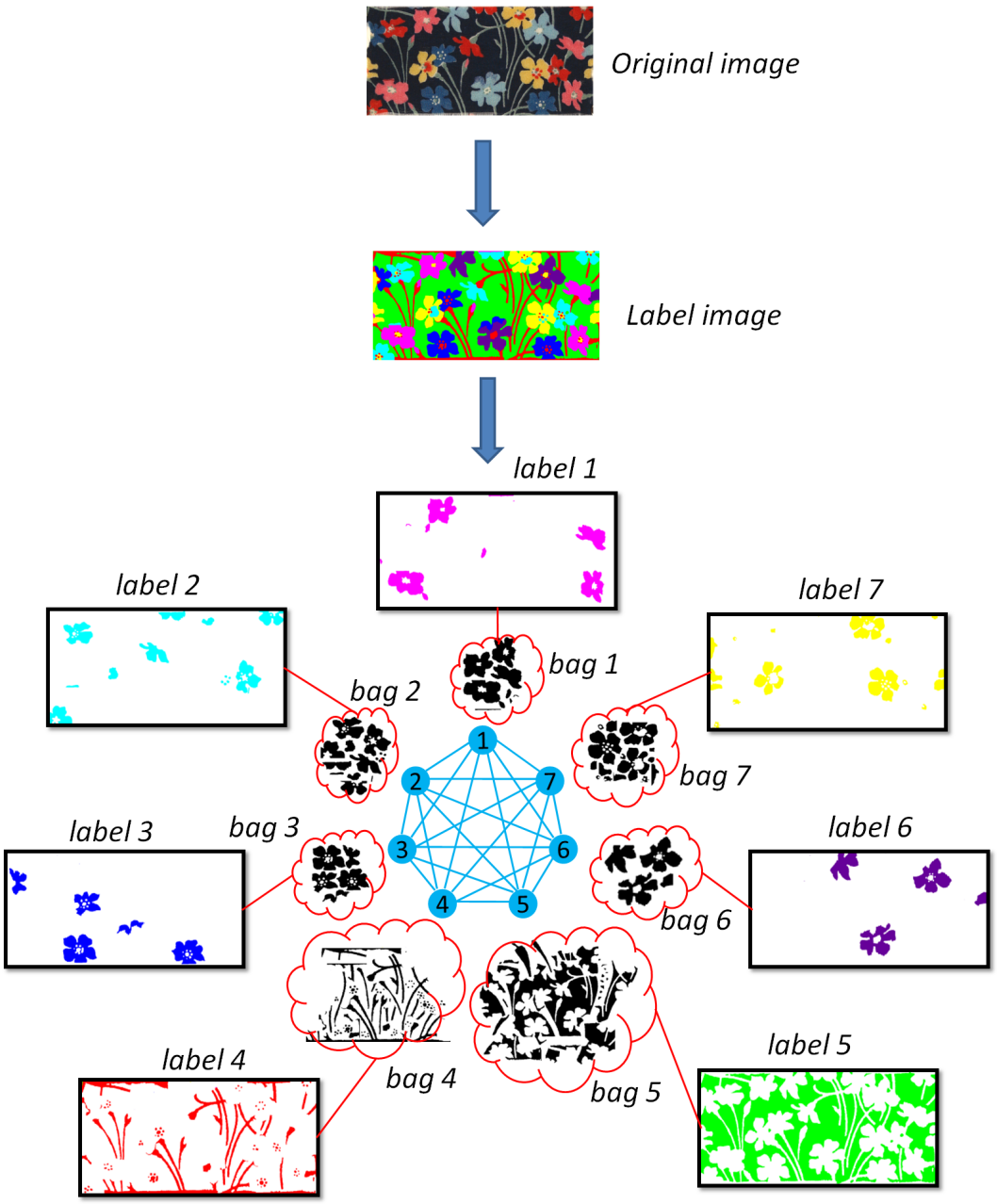


Figure 3.8: An original image is segmented into seven groups of regions, and each group of regions share the same label. The label image is shown in the centre. A weighted graph can be constructed in which each vertex is associated with a group of regions that share the same label. Each group of regions is represented as a bag of shapes. Edge weights encode relationships between the groups. (Image courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

extent to which the groups' regions are spatially adjacent, or the dissimilarity of their respective bags of shapes.

An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of vertices $\mathcal{V}$ and a set of edges $\mathcal{E}$. Each vertex $v \in \mathcal{V}$ is associated with a group of regions (bag of shapes). Two ways in which edge weights can be assigned are:

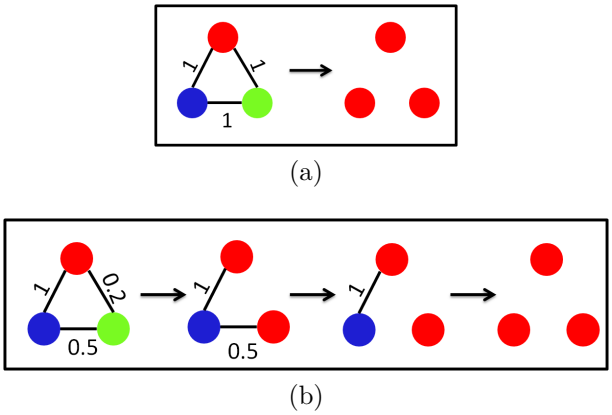
1. A weight is assigned the arc length of the common boundary shared by the groups of regions. This is the extent to which two groups' regions are spatially adjacent.

2. A weight is assigned a measure of dissimilarity of the bags of shapes. Each bag of shapes is described as a histogram of codewords, and the distance between pairs of histograms is calculated as the weight.

At the same time, the complement of a graph $\mathcal{G}$ can be constructed as well. The complement of a graph is the graph $\mathcal{G}'$ that has the same vertex set and an edge set consisting of the edges not present in $\mathcal{G}$ [112]. The graph sum $\mathcal{G} + \mathcal{G}'$ is the complete graph in which every pair of distinct vertices is connected by an edge. After constructing the graph and its complement, the graph feature can be extracted. The graph feature is extracted based on the chromatic number and domination number in graph theory. Section 3.4.2 introduces the chromatic number and domination number, and the corresponding graph features extracted.

3.4.2 Chromatic number and domination number sequences

In graph theory, the chromatic number of a graph, $\mathcal{G}$, is the smallest number of colours needed to colour the vertices without adjacent vertices sharing the same colour [112]. In general, a graph with chromatic number k is said to be a k-chromatic graph.



(a)

(b)

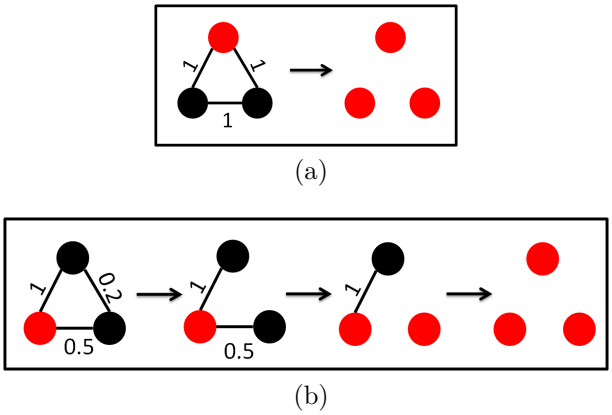
Figure 3.9: Graph minimal colouring sequences obtained by deleting edges in the order of weight.

Consider toy examples with three groups of regions as vertices of a graph, fully connected to each other. The edge weights are assigned values proportional to the arc lengths of the common boundaries shared by the groups of regions. Two such graphs are as follows.

1. Figure (3.9a) shows an example in which the groups of regions are equally adjacent to each other so that all edges are assigned the same weight. Deleting edges in order of weight generates the chromatic number sequence $3 \rightarrow 1$. This is an extreme case.

2. Figure (3.9b) shows an example in which edges are assigned different weights. Deleting edges in order of weight generates the chromatic number sequence $3 \rightarrow 2 \rightarrow 2 \rightarrow 1$.

These examples illustrate that deleting edges by weight results in sequences of chromatic numbers that depend on the adjacency relationships of the region groups. Similarly, sequences of chromatic numbers can be computed from the complement graphs. Calculating chromatic numbers is NP-complete [112] and no convenient method is known for determining the chromatic number of an arbitrary graph [50]. However, the graphs obtained have relatively few vertices so



(a)

(b)

Figure 3.10: Graph dominating set sequences (red dots) obtained by deleting edges in the order of weight.

they can be computed quickly. A graph colouring algorithm based on backtracking was used [32].

Another measure from graph theory that can reflect relationships between the vertices (groups of regions) is the domination number [52]. A dominating set of a graph is a subset $\mathcal{D}$ of $\mathcal{V}$ such that every vertex in $\mathcal{V} - \mathcal{D}$ is joined to at least one vertex in $\mathcal{D}$ by some edge. Figure 3.10 shows dominant sets (red nodes) of groups obtained by deleting edges in order of weight. The number of red nodes is the domination number. Thus, in the example shown in Figure 3.10 (a), the domination number sequence is $1 \rightarrow 3$ and in the example of 3.10(b) the domination number sequence is $1 \rightarrow 1 \rightarrow 2 \rightarrow 3$. Similarly, we can compute sequences of domination numbers from complement graphs. Finding the domination number is also an NP-complete problem [42]. The domination number was calculated using the Bron-Kerbosch algorithm [21].

The sequence of (normalised) weights of those edges whose removal changes the chromatic number constitutes a feature vector. Similarly, a feature vector can be computed based on changes in domination number. We call these feature vectors graph features. Figure 3.11 plots chromatic number against normalised weights for one of the examples above. The red squares corresponding to the edge weights which make the chromatic change, form the resulting graph feature vector $[1.0, 0.2, 0.0, \ldots, 0.0]$.

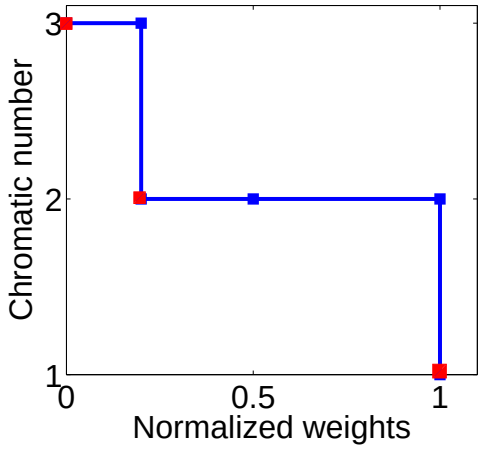


Figure 3.11: The change in chromatic number as edges are removed in order of weight for the example in Figure 3.9b. The resulting feature vector is $[1.0, 0.2, 0.0, \ldots, 0.0]$.

The algorithm of extracting graph features is shown in Table 3.1:

1. Construct a graph by assigning edge weights with dissimilarity or common arc length;
2. Sort the weights in descending order;
3. Obtain a sequence of graphs by deleting edges in descending order;
4. Obtain the graph feature by concatenating the edge weights which make the chromatic or domination number change.

Table 3.1: Graph feature extraction algorithm

As introduced in Section 3.4.1, two kinds of graphs are created and differ in the edge construction. The graph features extracted from the graph constructions in which the edges show the spatial coherence between groups of regions are denoted as G_{adj}, and G'_{adj} for the complements. The graph features obtained from the graph constructions in which the edges means the dissimilarity of pairs of groups regions are denoted as $G_{dis}$and G'_{dis} for the complements.

In summary, given a collection of shapes from training images, a codebook was calculated by running k-means [76] on the shape descriptors, GFD. Codewords are defined as the centres of the clusters [39]. A given shape can be assigned to the nearest codeword. A set of shapes can be represented as a histogram of the codewords. To capture the spatial structure information, graphs and their complements based on region labels are constructed. Corresponding graph features

can be extracted based on chromatic number and domination number in graph theory. Therefore, images can be described by a feature vector combining the shape feature (the histogram of codewords) and the graph features, which can be used for standard machine learning classifier, e.g. Support Vector Machine (SVM).

3.5 Summary

This chapter introduced the bag of shapes model. Firstly, literature review reflects the existence of research work about image representation using bag of words model, and the visual words in this kind of model are detected regions or uniform image patches. In Chapter 2, it is shown that reasonable segmented regions can be obtained from MRF pixel labelling. Therefore, we propose the bag of shapes model in which visual words are segmented regions from MRF pixel labelling and each single region is described using the popular and efficient Generic Fourier Descriptor (GFD). Without feature descriptor quantisation, image features are matrices of $s \times 48$, where s is the number of shapes. This representation will be applied to the task of textile design classification in Chapter 4 using Naive Bayes Nearest Neighbour classification. This representation will be investigated and evaluated through comparison with bags of SIFT features.

One of the main drawbacks of bag of words models is the lack of spatial information between visual words. We construct graphs and their complements in order to capture such spatial information. Regions with the same label are treated as a group and each group is associated uniquely with a vertex in an undirected, weighted graph. Each region group is represented as a bag of shape descriptors. Edges in the graph denote either the extent to which the groups' regions are spatially adjacent or the dissimilarity of their respective bags of shapes. It is

tractable to use such graphs because the number of the vertices is not big. Once the graphs are constructed , their complements can be obtained directly. Series of unweighted graphs can be generated from the weighted graphs via edge removal. For the series of graphs, the corresponding sequences of chromatic number and domination number can be obtained. Those edge weights which make the chromatic number and domination number change are considered as graph features. The linear SVM classifier will be used to verify the effectiveness of the representation and the graph feature in the application of textile design classification in Chapter 4.

Chapter 4

Textile design classification

4.1 Introduction

Chapter 3 provides bags of shapes image representation based on MRF segmentation regions, and the spatial information are extracted by constructing region label graphs using MRF labels. The representation and features can be used to compare and match images that is helpful in many computer vision applications, such as image classification, retrieval, browsing, and tagging. To investigate the use of such a representation and features, this Chapter will apply the representation and features to an image classification task. This task can be helpful for access and management of digital images.

Images for the experiments are textile images (to be introduced in Section 4.2.3.1). Textile image classification is to assign each textile image a text descriptor based on the design type. This can help designers to access the database more easily and quickly, which contributes to design efficiency improvement, and greater potential for increasing revenue.

Classification experiments using bag of shapes models are reported in Section 4.2. This compares the use of bags of shapes extracted from MRF region seg-

mentation with the use of bags of SIFT descriptors. The comparison is in terms of classification accuracy and computational efficiency. This comparison is performed using a simple but effective classification scheme based on Naive Bayes Nearest Neighbour (NBNN) searches in training data available for each class [14]. This scheme was adopted because it has obtained state-of-the-art performance on several well-known natural scene data sets using features such as SIFT [14]. It is suitable for bag of shapes models because it involves neither descriptor quantisation nor parameter-sensitive learning algorithms.

Classification experiments using region label graphs are reported in Section 4.3. Feature vectors combine graph features obtained from constructing region label graphs and shape features calculated as histograms of shape descriptor codewords. They compare different feature sets based on a linear SVM classifier [23].

The remainder of this chapter is structured as follows. Section 4.2 overviews the NBNN classification method, provides details of experiments using bag of shapes models, and finally draws some conclusions. Section 4.3 provides details of experiments using the representation of region label graphs based on a linear SVM classifier and draws some conclusions.

4.2 Classification using bags of shapes

As introduced in Chapter 3, images can be represented using bag of shapes models. Without feature quantisation and codebook calculation, images are described as feature matrices directly. Simple but effective non-parametric nearest neighbour based classification is used for this kind of representation, which needs no descriptor quantisation.

4.2.1 Nearest-Neighbour based classifier

The nearest neighbour classifier has been a ubiquitous and workhorse classifier [5, 144, 28]. Given a query vector and a set of labelled instances, the task of the classifier is to predict the class label of the query vector on the predefined classes. The $K-$Nearest neighbour classification algorithm attempts to find the neareast neighbours of the query vector and uses a majority vote to determine the class label of it.

Boiman *et al.* [14] pointed out that feature quantisation commonly used in image classification results in the degradation of neareast neighbour classifiers. Feature quantisation is usually used to obtain compact image representation. A large collection of feature descriptors (usually thousands of feature descriptors from training data) is quantised into a relatively smaller collection of descriptors (typically $100 - 1000$ representative descriptors). It is known that highly frequent descriptors have lower quantisation error, while low frequency descriptors result in high quantisation error. However, the most frequent descriptors are usually rich in all of the classes, but discriminative feature descriptors (some informative descriptors) only appear a lot in one (or few other) class, but rarely appear in other classes. Therefore, the discriminative feature descriptors tend to be few in the database which leads to higher quantisation error. Besides, when dividing densely sampled descriptors into bins, the bin density follows a long tail (or heavy tail) distribution [59, 123]. It means that most descriptors are infrequent, then isolated in the descriptor space. Therefore the long-tail distribution descriptors are even more inappropriate for quantisation. Boiman *et al.* [14] run experiments on the effects of descriptor quantisation and show quantitatively that quantisation error leads to the degradation of discriminative power of feature descriptors, and the more informative the feature descriptor is, the more severe the degradation in discriminative power. They propose a Naive Bayes Nearest Neighbour classifier which requires no feature quantisation, and this will be introduced in

Section 4.2.2.

4.2.2 Optimal Naive Bayes classifier estimation

A query image, Q, can be classified by assigning it to the class, $\hat{C}$, with largest posterior probability: $\hat{C} = \arg\max_C p(C|Q)$. This is equivalent to maximising the class likelihood if a uniform prior is adopted: $\hat{C} = \arg\max_C p(Q|C)$. Given a representation of Q in terms of a set of descriptors $\mathbf{d}_1, ...\mathbf{d}_n$ assumed to be i.i.d.(naive Bayes assumption). The classification rule becomes:

$$\hat{C} = \arg\max_C \sum_{i=1}^{n} \log(p(\mathbf{d}_i|C)) \tag{4.1}$$

The optimal classifier under the naive-Bayes assumption in Equation (4.1) needs to calculate the probability density for every descriptor. Since the number of descriptors in a training set will be very large, a Parzen density estimation is used to estimate the continuous descriptor probability density $p(\mathbf{d}|C)$ [37]:

$$\hat{p}(\mathbf{d}|C) = \frac{1}{M} \sum_{j=1}^{M} K(\mathbf{d} - \mathbf{d}_j^C) \tag{4.2}$$

where M is the number of descriptors obtained from all of the training images of class C and K is a Parzen kernel function (which is non-negative and integrates to 1). In order to estimate Equation (4.2) with high accuracy, all of the descriptors should be used in principle. But computing the sum over M terms is expensive, since it needs to calculate $\mathbf{d} - \mathbf{d}_j^C (j = 1, ..., M)$ for every descriptor in each class. However, this Pazen estimator in Equation (4.2) is well approximated using a sum only over l nearest neighbours because of the long-tail properties of the descriptor distribution:

$$p_{NN}(\mathbf{d}|C) = \frac{1}{M} \sum_{j=1}^{l} K(\mathbf{d} - \mathbf{d}_{NN_j}^C) \tag{4.3}$$

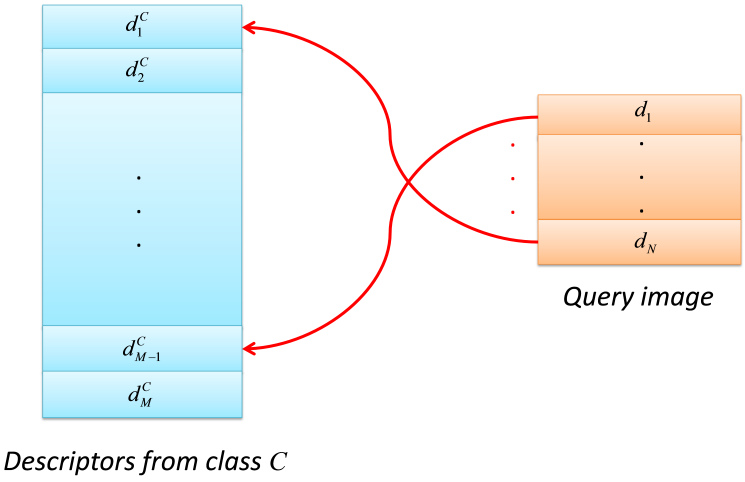


Figure 4.1: Each descriptor in the query image matches to the nearest neighbour descriptors in class C. The descriptors in class C is the collection of all of the descriptors from training images in this class.

Boiman *et al.* [14] provided empirical evidence using several popular image classification data sets that even using a single nearest neighbour ($l = 1$) can be effective and indeed highly competitive with the state-of-the-art in image classification. They observed very little effect on the discriminative ability of local feature descriptors. Therefore, in the case of a Gaussian kernel, $K_g(\mathbf{d} - \mathbf{d}_{NN_j}^C) = \frac{1}{(2\pi)^{d/2}\sigma^d} \exp(\frac{-||\mathbf{d}-\mathbf{d}_{NN_j}^C||^2}{2\sigma^2})$,

$$log(p(Q|C)) \propto -\sum_{i=1}^{n} ||\mathbf{d}_i - NN_C(\mathbf{d}_i)||^2$$

This results in the following simple Naive Bayes Nearest Neighbour (NBNN) algorithm [14] as shown in Figure 4.1:

1. Calculate descriptors $\mathbf{d}_1, ...\mathbf{d}_n$ for a query image Q and for each such descriptor $\mathbf{d}_i$, find its nearest neighbour $NN_C(\mathbf{d}_i)$ for each class C.

2. Assign the query to the class

$$\hat{C} = \arg\min_C \sum_{i=1}^{n} \|\mathbf{d}_i - NN_C(\mathbf{d}_i)\|^2 . \tag{4.4}$$

In order to speed up nearest neighbour search, KD-trees were used. The time complexity is $O(N \log N)$, where N is the number of items to search [92]. We also experimented with an exponential kernel instead of a Gaussian:

$$K_e(\mathbf{d} - \mathbf{d}_j^C) = a \exp(-a||\mathbf{d} - \mathbf{d}_j^C||) \tag{4.5}$$

resulting in the classification rule:

$$\hat{C} = \arg\min_C \sum_{i=1}^{n} \|\mathbf{d}_i - NN_C(\mathbf{d}_i)\| . \tag{4.6}$$

This kernel has heavier tails. Additionally, we experimented with the L_1 norm in place of the L_2 norm since it is known to perform better for some high-dimensional matching problems (see e.g.[1]).

In summary, under the Naive Bayes assumption, the optimal image classifier can be estimated by NBNN. NBNN avoids descriptor quantisation and employs nearest neighbour distance in the space of the local image descriptors rather than in the space of images. This classifier is simple but its performance ranks among the top learning-based image classifiers [14]. In Section 4.2.3, classification experiments will be applied based on the NBNN scheme.

4.2.3 Experiments and results

This section describes the experimental results of image classification using bags of shapes and also the comparison with bag of SIFT features based on NBNN. Section 4.2.3.1 introduces the experimental data first, and then Section 4.2.3.2 describes the method to obtain local regions/patches and representation in both bag of shapes and bag of SIFT features. Experiment details are provided in Section 4.2.3.3 and Section 4.2.3.4 shows some results.

Figure 4.2: Textile images of seven classes manually classified by experts. (Image courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

4.2.3.1 Experimental data

Images were from the commercial archive of Liberty Art Fabrics, which holds more than 10,000 images, mainly of textile images. There are 5015 images in the database having information of number of colour dyes used in images (MRF segmentation needs this information) and these images are labelled for experiments. In order to test the effectiveness of the image representation and features, each image was assigned a text descriptor by Dr. Ward who has over 25 years

experience as an academic in textiles, clothing, and design. Seven text descriptors were generally termed "style" within the archive database: floral (949 images), paisley (84 images), stripe (108 images), leaf (256 images), geometric (202 images), spot (131 images), and check (135 images). These text descriptors were used to compare human labelling with machine classification. Therefore, these images were categorised into seven classes according to the text descriptors. We randomly chose 70 from each class to make sure each class had the same number of images. So 490 images (7 classes, 70 per class) were used in the experiments. The average size of each image is 500×500. Figure 4.2 shows some examples from the seven classes. It should be noted, however, that one of the descriptors (geometric) defined a broad design style that included images with more variable content but with geometric attributes (e.g., angles, lines, simple shapes) as shown in Figure 4.3(a). Figure 4.3(b) is an image assigned with descriptor “floral” by its design type, but obviously, flower is not the only content contained in the image; some other elements, leaf, fruit (circle-like) are included. This may result in confusion. Some terms (floral, paisley, and leaf) referred to the design content of the image in familiar textile design terms (especially in the cases of floral and paisley); others (stripe, spot, and check) provided both a description of style and visual content. All of the elements make textile image classification challenging. These varied ways of describing the images should be further refined and developed in future work.

4.2.3.2 Local region detection and representation

We compared bag of shapes with bag of SIFT features using the classification methods described in Section 4.2.1. For both of the cases, the first step is to detect the regions and describe them using feature descriptors.

(a)

(b)

Figure 4.3: (a) Image examples from geometric class. (b) A floral image. (Image courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

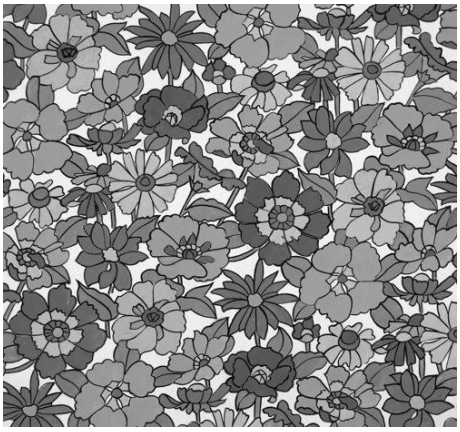
- Bag of shapes: Images were represented by collections of segmented regions. MRF pixel labelling introduced in Chapter 2 was used to obtain the label images and the segmented regions were connected components of the label images. Regions smaller than 50 pixels in area were ignored. The number of regions obtained varied from image to image, averaging around 500. For efficiency, instead of using all regions, only the largest regions in an image were used to represent it. As well as reducing computational expense, this reflected the intuition that larger regions are more likely to be informative for classification. Specifically, the obtained regions of an image were sorted in ascending order, and in each experiment an upper bound n_r was set on the number of regions per image. The top n_r regions in the sorted list were used for matching. If the number of regions in an image was below this bound, all its regions were used. Regions were described using 48−dimensional GFD features.

- Bag of SIFT features: Images were represented by collections of local patches and SIFT [80] was used to describe each local patch. SIFT features are a very popular local descriptors, which are histograms of gradient location and orientation. The location is quantised into 4 blocks and gradient angle is quantised into eight orientations. Therefore, 128−dimensional SIFT features can be obtained. Two methods of computing bags of SIFT features were used:

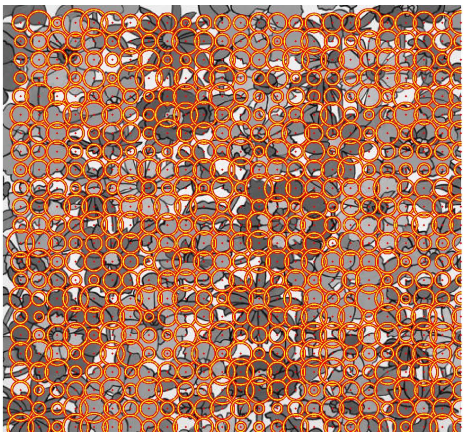
  1. EP-SIFT: Local patches were horizontally and vertically separated by 20 pixels (as shown in Figure 4.4(b)).

  2. RP-SIFT: 500 local patches were sampled at random (as shown in Figure 4.4(c)).

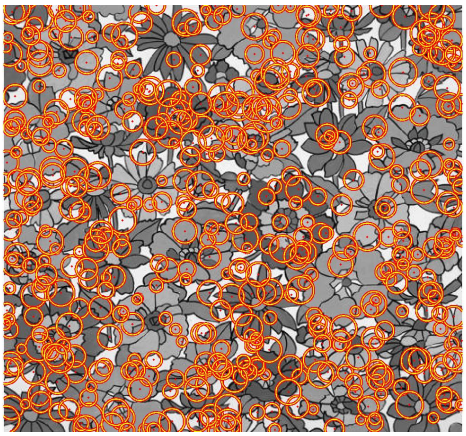
In both cases, patch diameters were sampled at random in the range 10 to



(a)



(b)



(c)

Figure 4.4: (a) Original grey scale image (b) Evenly sampled image patches (c) Randomly sampled image patches (Image courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

30 pixels. Each patch was described using a 128-dimensional SIFT feature vector[89].

4.2.3.3 Experiment set-up

As benchmarking procedure, the image set for each class was divided at random into $n_{training}$ training images and n_{test} test images with $n_{training} + n_{test} = 70$. A collection of feature descriptors for each class was obtained from the training images. When classifying a new test image, the decision was made to the class label that gives the highest likelihood, equally the lowest distance

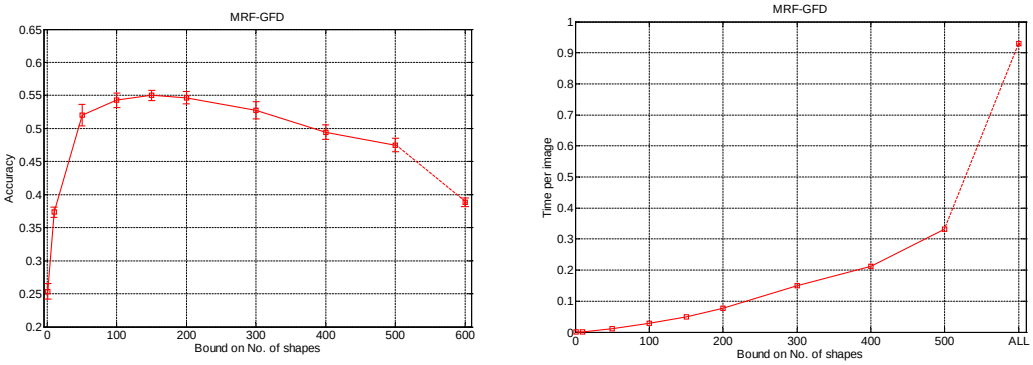
in Equation (4.5) and (4.6) depending on the kernel. We randomly selected $n_{training} = 1, 10, 20, 30, 40, 50$ images per class. The entire procedure was repeated several times (selecting the training and testing images randomly), and the classification performance each time is the mean of classification rate per class. A confusion matrix was used to illustrate the performance of the classification. The x-axis represents the model for each class of images and the y-axis represents the ground truth classes of the images. Therefore the ideal case is expected to be a completely white diagonal line to show the discriminative power of the model to classify images.

Experiments applied were: (i) Test bag of shapes (MRF-GFD) (MRF segmented regions described using region shape descriptor GFD) and compare it with bag of SIFT features using NBNN. (ii) Find the appropriate number of descriptors in both bag of shapes (MRF-GDF) and bag of SIFT features (RP-SIFT). (iii) Compare different kernel functions (Gaussian kernel and exponential kernel) and different distances (L_1 and L_2) for classification.

4.2.3.4 Results

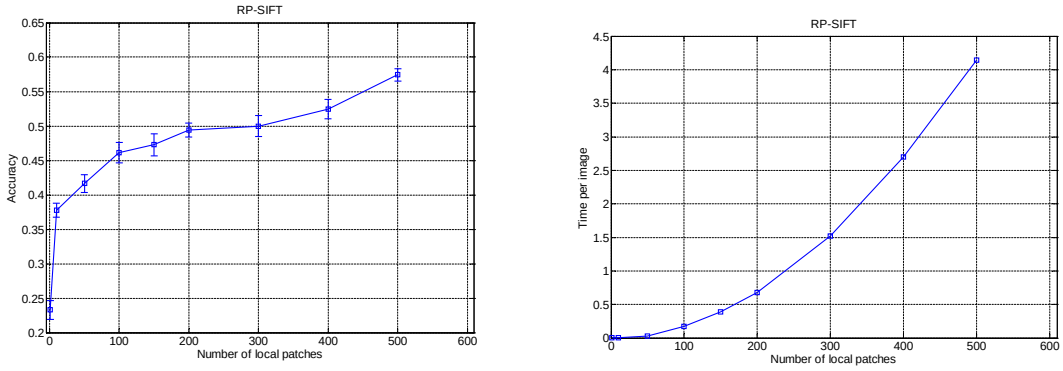
Figure 4.5 shows how classification accuracy and average matching time varied with the bound n_r on the number of descriptors per image. When $n_r = 150$, the accuracy reaches a peak, but after that, the accuracy reduces gradually. When using all of the shapes (see Figure 4.5(a)). This shows that small regions may not help to improve the classification accuracy and validate our intuition that large regions are helpful and informative. Figure 4.6 shows how classification accuracy and average matching time vary with the number of sampled local patches noted as n_s in the case of SIFT features (RP-SIFT). With the increase of n_s , the accuracy is improved (see Figure 4.6(a)). The number of training examples per class was $n_{training} = 50$.

Figure 4.7 illustrates the effect of using different kernels (Gaussian K_g and ex-



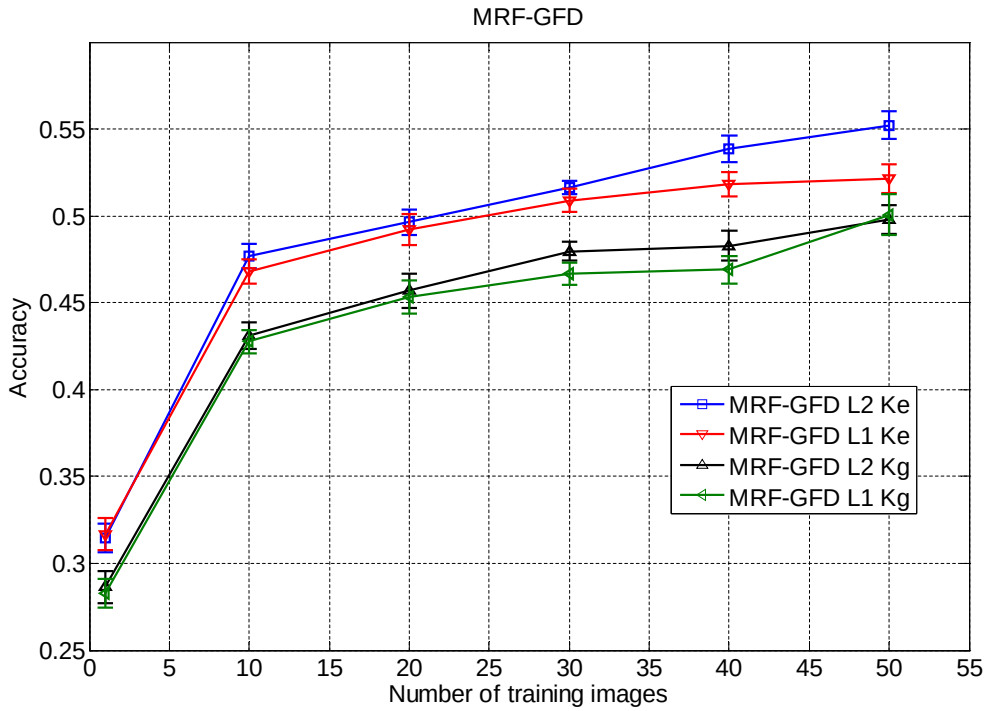
(a) (b)

Figure 4.5: (a) Classification accuracy using bags of shapes model (MRF-GFD) against bound on number of shapes. The error bars denote ± standard error of running 10 times for $n_{training} = 50$. The crossses denote mean of 10 times classification accurate rate. (b) Average executive time against bound on number of shapes.

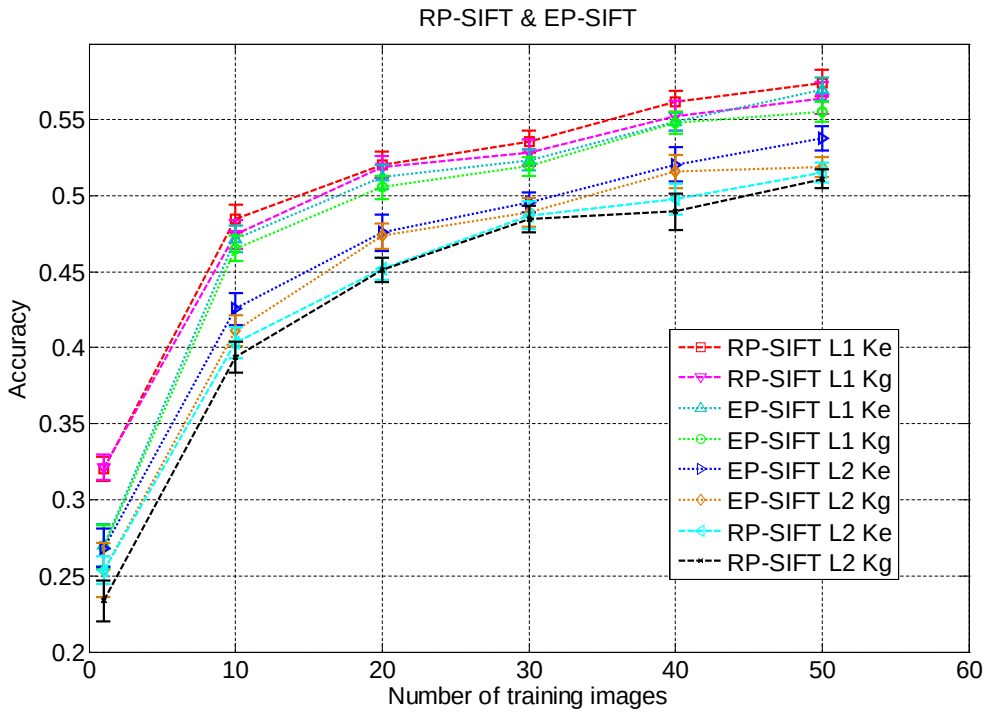


(a) (b)

Figure 4.6: (a) Classification accuracy using bags of SIFT feature with local patches sampling randomly (RP-SIFT) against number of local patches. The error bars denote ± standard error of running 10 times for $n_{training} = 50$. The crossses denote mean of 10 times classification accurate rate. (b) Average executive time against number of local patches.



(a)



(b)

Figure 4.7: Classification accuracy. The error bars denote ± standard error of running 10 times for one fixed number of training images. The crosses denote mean of 10 times classification accurate rates.

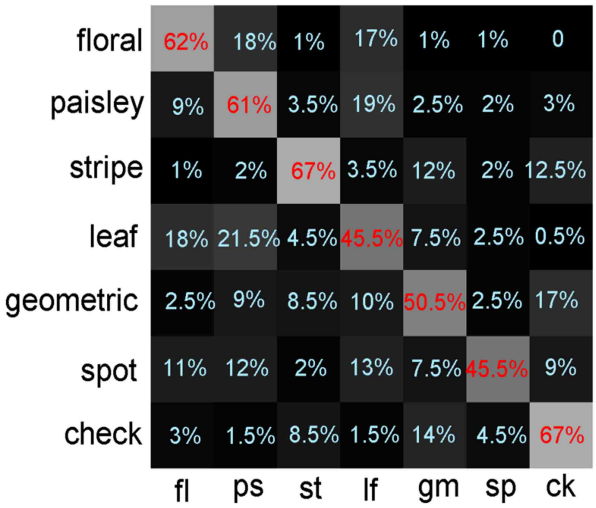


Figure 4.8: An example confusion matrix using bags of shapes for classification

ponential K_e) and different norms (L_1 and L_2) for classification. The results are plotted for training set sizes $n_{training} \in \{1, 10, 20, 30, 40, 50\}$. 500 descriptors in the case of RP-SIFT, and an upper bound of 150 descriptors in the case of MRF-GFD were used respectively. Each point on the plot represents an average of 10 trials with randomly selected training examples. For both descriptor types, the exponential kernel gave better accuracy than the Gaussian kernel, irrespective of the norm used. MRF-GFD had slightly better results using L_2 than L_1. In contrast, L_1 gave better results than L_2 with the SIFT features, since SIFT feature has higher dimensionality and L_1usually performs better in higher dimensional matching.

Figure 4.8 shows a confusion matrix obtained using the bag-of-shapes (MRF-GFD) method with $n_{training} = 50$. Images from leaf, geometric, and spot have lower classification accuracy. The ambiguous classes with leaf class mainly happen in paisley and floral. It is probably because images from leaf and paisley class have similar shapes, and images from floral usually contain leaves. Geometric class sometimes include the elements of square, so it is ambiguous with check, but check is often considered as geometric. Images from spot have almost the same mis-classification error with other classes except stripe class maybe because the shapes of spot appear in every class.

Using 50 training images per category it took on average 4.14s to compare one image with one class using 500 SIFT descriptors per image. Bag-of-shapes took 0.04s with up to 150 regions. When up to 500 regions were used it took 0.35s. Bag-of-shapes obtained similar accuracy to SIFT (Figure 4.7) at reduced computational expense. In the above experiment, matching was two orders of magnitude faster using bags of shapes than when bags of SIFT features were used. The shape features were of lower dimensionality than SIFT and there were fewer of them per image.

4.2.4 Discussion and conclusion

The bag of shapes model is a collection of shape descriptors of segmented regions from MRF. The number of regions used to represent an image was decided empirically, and a bound number of regions was set to improve the efficiency. A simple yet competitive classification scheme based on nearest neighbour matching was used. The method was compared to the use of bags of SIFT features and accuracy was comparable. The bag of shapes model had similar classification accuracy to bags of SIFT features. Two different kernels, Gaussian kernel and exponential kernel were used. In both representations, the exponential kernel resulted in better accuracy. To calculate distance between descriptors, L_1 and L_2 norms were used. In the case of GFD, L_1 and L_2 had similar performance, but L_2 was slightly better than L_1. Conversely in the case of SIFT, L_1 performs better since SIFT feature has higher dimension, and $L_1$is better for high dimensionality matching problems. Once features had been computed, matching using bags of shapes was considerably faster.

The experimental database was very challenging. The intra-class variation is very large, and some of the classes have common elements. The average classification accuracy of bags of shapes model is about 55% , and is about 58% for bag of SIFT model. Although the accuracy is not very high due to the challenging dataset,

using bags of shapes model can achieve similar performance to the popular local feature SIFT. Because of the lower dimensionality of GFD, matching is two orders of magnitude faster than using SIFT, so bags of shapes gain the advantage of time efficiency.

One possible reason that degrades the classification performance using bags of feature models is that they lack the spatial information of images. Bag of shapes model uses MRF pixel labelling result to get the segmented regions, but it ignores the labels that may be helpful to extract some spatial features. Section 3 introduced another novel representation, region label graphs, which can be used to capture the spatial information of images. Experimental results on textile design classification using region label graphs will be reported in Section 4.3.

4.3 Classification using region label graphs

This section reports the experimental results of classification using region label graphs as described in Chapter 3. Recall that the image features obtained form a compact feature vector containing two parts: shape and graph features. Since this feature space is of fixed dimensionality, standard machine learning methods can be used. Many learning based classification methods have been used [68, 39, 125, 145]. The purpose here is to investigate the effectiveness of the region label graphs' representation and the extracted graph features, so linear SVM classifier [23] was used because it is a fast machine learning algorithm for solving multi-class classification. More complicated classifiers can be tested in the future that may help to improve the classification accuracy.

A comparison of approaches to handle the multi-class case as opposed to the binary case is presented in [53]. The strategy in LibSVM [23] for multi-class classification is to build a set of one-versus-one classifiers, and choose the class that is selected by the most classifiers. The binary SVM separates a given set of

binary labelled training data with a hyper-plane that is maximally distant from them.

Given training vectors $\mathbf{x}_i, i = 1, ...L$, ($L$ is the number of the training vectors), in two classes, and a vector $y_i \in (-1, 1)$, the binary standard SVM is to solve a quadratic optimisation problem:

$$\min_{\mathrm{w},b,\xi} \frac{1}{2}\mathrm{w}^T\mathrm{w} + C_p \sum_{i=1}^{L} \xi_i \tag{4.7}$$

subject to $y_i(w^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i$, and $\xi_i \geq 0, i = 1, ...L$.

$C_p > 0$ is the penalty parameter of the error term. The decision function of any testing vector is:

$$f(x) = sign[\mathrm{w}^T\phi(\mathbf{x}) + b] \tag{4.8}$$

4.3.1 Experiments and results

An experiment was set up to investigate the effectiveness of graph features obtained by constructing region label graphs in the application of textile design classification. This experiment used the same dataset as in Section 4.2.3.1. Section 4.3.1.1 describes codebook formation and Section 4.3.1.2 introduces the implementation of graph feature extraction. Experimental details and results will be provided in Section 4.3.1.3 and Section 4.3.1.4.

4.3.1.1 Codebook formation

Images were represented by bag of shapes or bag of SIFT features respectively. Similarly to Section 4.2.3.2, in the case of bag of shapes, each image was represented by a collection of MRF segmented regions and each region was described

using a 48−dimensional Generic Fourier Descriptor. In the case of bag of SIFT, we use RP-SIFT described in Section 4.2.3.2, 500 local patches were randomly sampled from each image, and each patch was described as a 128−dimensional SIFT feature vector using an existing implementation [89]. For both cases, given the collection of region descriptors from training images of all classes, we learned the codebook using the k-means algorithm [39], centres of the clusters were the codewords, and images were represented by k dimensional histograms of the codewords. $k = 500$ was chosen empirically in both cases. To distinguish this feature (histogram of codewords) from graph features (to be introduced in Section 4.3.1.2), we call histogram of GFD codewords 'GFD feature' in the case of bag of shapes, and histogram of SIFT codewords 'SIFT feature' in the case of bag of SIFT.

4.3.1.2 Graph feature extraction

There are two ways to encode spatial information on the edges of graphs: For each pair of groups' regions, the edge is (1) the arc length of common boundary between pairs of groups' regions, normalised by the maximum number of boundary pixels, or (2) the dissimilarity between groups of regions normalised by the maximum dissimilarity value. In the case of using dissimilarity between groups' regions, L_2 distance is used as the distance between codeword histograms of pairs' groups. The graph feature is extracted by calculating chromatic number and domination number. Chromatic numbers were calculated using the TORSCHE Scheduling Toolbox [72]. The implementation for calculating domination numbers was based on the Matgraph toolbox[108]. Figures 4.9(a) and 4.9(c) plot the chromatic numbers and domination numbers of a graph and its complement against the weights of edges removed. Note that these are monotonic functions and that the weights are normalised in the range $[0, 1]$. 12-dimensional feature vectors were derived in the case of both chromatic numbers and domination numbers, since the number of colour dyes contained in a textile design is usually no

more than 12. The corresponding feature vectors are shown in 4.9(b) and 4.9(d).

G_{dis} denotes the graph features obtained by constructing graphs using dissimilarity as edge weights, G'_{dis} denotes features obtained from the complement graphs, and $G_{dis} + G'_{dis}$ denotes the combination of features from graphs and complements using dissimilarity as edges. G_{adj} denotes the graph features obtained by constructing graphs using arc length of the common boundary shared by the groups of regions, G'_{adj} denotes features obtained from the complement graphs, and $G_{adj} + G'_{adj}$ denotes features obtained from graphs and complements using arc length of common boundary as edges. There are many other combinations of these features, we also try the combination of G_{adj} and G_{dis}.

4.3.1.3 Experiment set-up

Images used in experiments were the same as in Section 4.2.3.1: 490 images, 7 classes, 70 per class. k-fold cross validation was used to evaluate the accuracy of image classification. Images in each class were randomly partitioned into k sub-samples, and of the k sub-samples, a single sub-sample was validation data for testing, and the rest was training data. The process was repeated $k = 7$ times, and for each fold, it was run ten times. The accuracy was the average of the seventy runs. A multi-SVM classification method based on one-versus-one binary linear SVM classifier [23] was used. There are seven classes of textile images, therefore there are $\frac{n(n-1)}{2} = 21(n = 7)$ classifiers. When classifying a new test image, the decision is made as the class label that is selected by the most classifiers. The penalty parameter C which penalises the error term in Equation (4.7) was set as default value 0.5.

The experiments applied were:

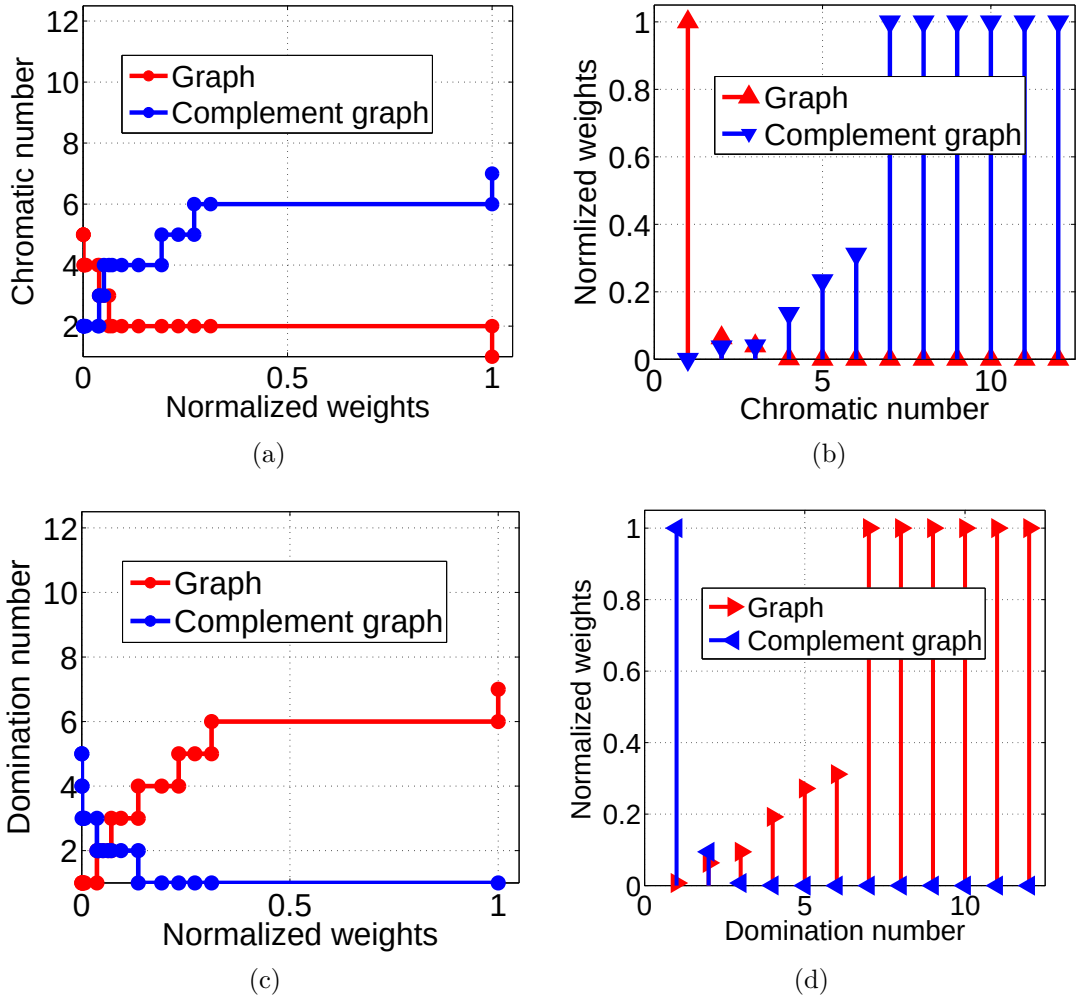
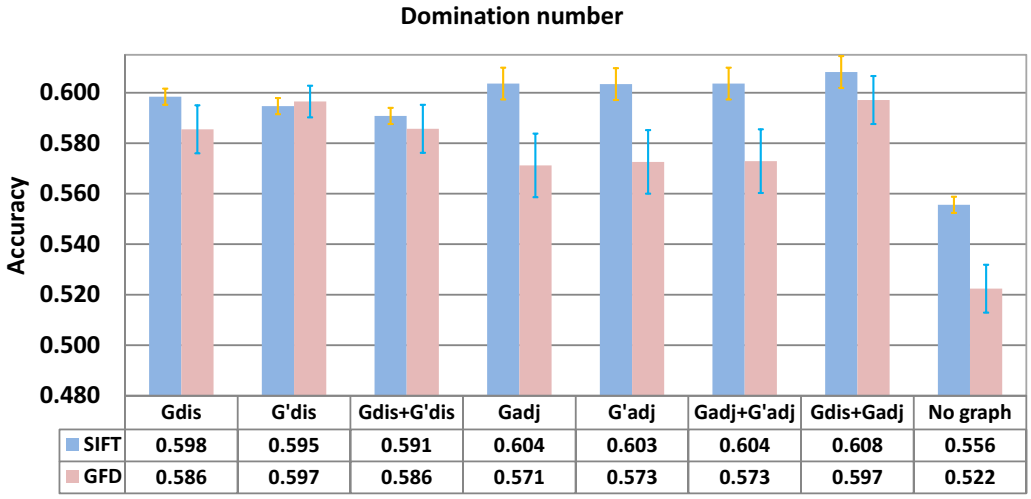


Figure 4.9: (a) Chromatic numbers of G and G' plotted against normalised edge weights. (b) Features derived from chromatic numbers. (c) Domination numbers of G and G' plotted against normalised edge weights. (d) Features derived from domination number.

- Test graph features obtained from calculating chromatic number and domination number respectively for textile design classification.
- Compare classification accuracy in both bag of shapes and bag of SIFT features models. The features of an image were:
  - GFD/SIFT feature only.
  - Combination of GFD/SIFT feature and graph features. Graph features were from different graphs and their complement construction:
    * Graph and complement constructed using the dissimilarity of groups' of regions: G_{dis}, G'_{dis}, $G_{dis} + G'_{dis}$
    * Graph and complement constructed using the arc length of common boundary between groups' of regions: G_{adj}, G'_{adj}, and $G_{adj} + G'_{adj}$

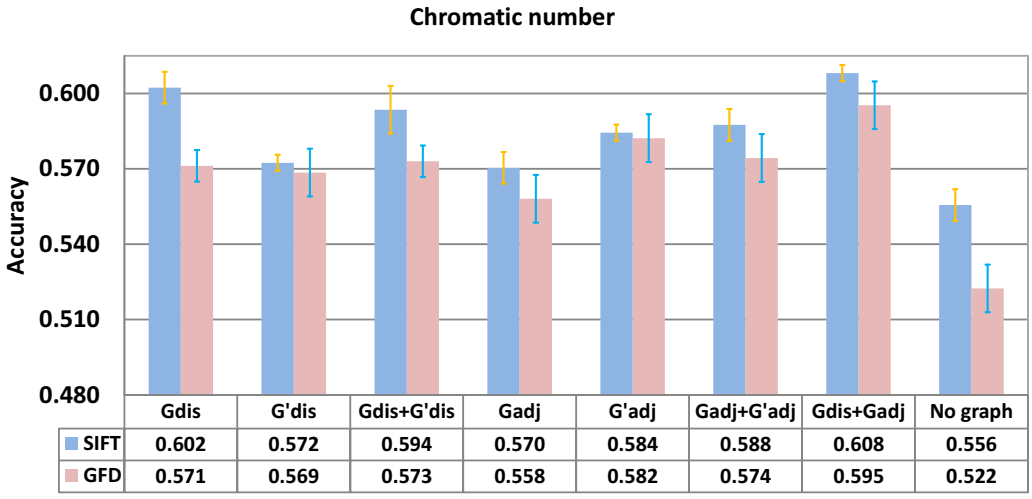
4.3.1.4 Results

The accuracy of classification using different feature sets is shown in Figure 4.10 and 4.11. The graph features used in Figures 4.10 and 4.11 were calculated from domination numbers and chromatic numbers respectively. Error bars denote ± one standard error estimated over 10 runs of seven-fold cross validation. No matter what kind of graph features are used and in both cases of using GFD feature and SIFT, the results suggest that classification accuracy is better than using GFD or SIFT features alone. In general, SIFT appears to be slightly more accurate than GFD. Classification can achieve the accuracy of 52.2% by using shape feature only, and 55.6% by using SIFT only. This suggests that using



	Gdis	G'dis	Gdis+G'dis	Gadj	G'adj	Gadj+G'adj	Gdis+Gadj	No graph
SIFT	0.598	0.595	0.591	0.604	0.603	0.604	0.608	0.556
GFD	0.586	0.597	0.586	0.571	0.573	0.573	0.597	0.522

Figure 4.10: Comparison classification results of different feature sets. The graph features are obtained by calculating domination number.



	Gdis	G'dis	Gdis+G'dis	Gadj	G'adj	Gadj+G'adj	Gdis+Gadj	No graph
SIFT	0.602	0.572	0.594	0.570	0.584	0.588	0.608	0.556
GFD	0.571	0.569	0.573	0.558	0.582	0.574	0.595	0.522

Figure 4.11: Comparison classification results of different feature sets. The graph features are obtained by calculating chromatic number.

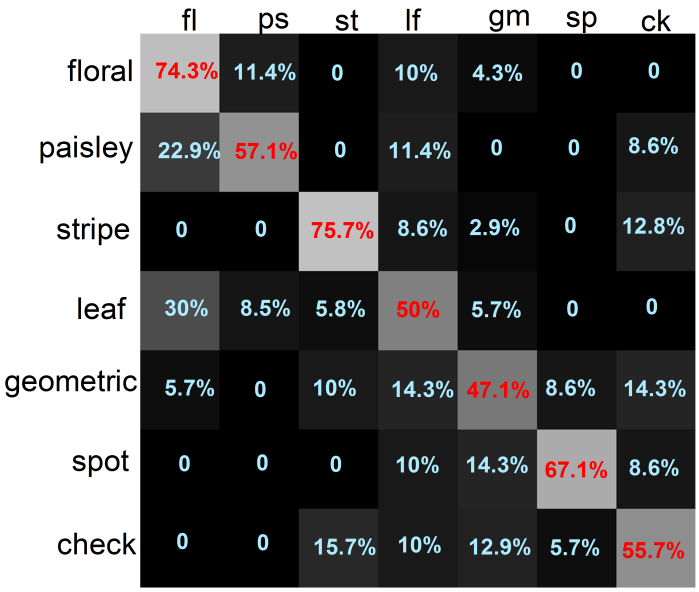


Figure 4.12: A confusion matrix obtained using shape feature and graph feature(G_{dis})

bags of shapes model based on NBNN framework performs better (GFD 55% and SIFT 58%) than using linear SVM without spatial information. However, based on linear SVM, with the help of graph features, the accuracy of linear SVM classifier improves to [55.8%, 59.7%] using GFD and improves to [57% , 60.8%] using SIFT. Graph features help much more on GFD than SIFT to improve the classification accuracy. Graph features extracted by calculation the domination number generally perform slightly better than that of chromatic number. The combined use of graphs with edges representing adjacency and dissimilarity gave classification accuracy at least as good as other feature sets.

Figure 4.12 shows a confusion matrix obtained using GFDs and G_{dis} graph features derived from domination number. We can see that the floral class (74.3%) and stripe class (75.7%) can achieve relatively good classification accuracy. However the geometric class is still confused with other classes, e.g. checks, because of the large variation in this class.

Figure 4.13: Example images from the seven categories. Images on the left were correctly classified. Images on the right were sometimes misclassified. The wrongly assigned class labels are shown below these images. (Image courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

4.3.2 Discussion and conclusion

A multi-SVM classification method based on one-versus-one binary SVM classifier was used for textile design classification. A linear function was chosen as the kernel function due to its efficiency. 500 codewords for both GFD and SIFT were calculated respectively. The experimental results have demonstrated that an improvement in classification accuracy was obtained by combining GFD or SIFT features (histogram of codewords) with the proposed graph-based feature vectors based on a multi-linear SVM classifier. In the case of combining GFD features with graph features, the classification accuracy increased by 3.4% ~ 7.5%, and in the case of combining SIFT features with graph features, the classification accuracy increased by 1.4%~5.2%, depending on what graph features were combined. The graph features are obtained by using chromatic number and domination number. Generally speaking, in both case of GFD or SIFT features, features combined with graphs features obtained from domination number performed slightly better than that from chromatic number (not significantly). Overall, classification using SIFT was slightly better than using GFD based on linear SVM classifier, and classification combining SIFT with graph feature sets performed slightly better than combining GFD with the same graph feature sets.

Compared with bags of shapes or bags of SIFT models based on the NBNN framework (experiments in Section 3.3), the models based on linear SVM classifier without using graphs features can have similar classification accuracy. However they can achieve better results when combined with graph features. Some classes, like floral and stripe, can get the accuracy of approximately 75%, although the geometric class is still confused with other classes.

Chapter 5

Conclusions and future directions

This chapter provides, in Section 5.1, a summary of the conclusions drawn from the experiments carried out in the present work. In Section 5.2, potential future directions of work are discussed. Both sections are divided into subsections associated with the textile design segmentation, representation, and classification.

5.1 Summary and conclusions

5.1.1 Textile design segmentation

Chapter 2 overviewed the related works about image segmentation. Segmentation is an ill-defined problem and there is not a unified solution for it. Due to the properties of textile design images, and motivated by textile production, a textile design segmentation problem was defined as extraction of printed design and woven patterns, and was modelled as a pixel labelling problem.

An MRF multi-pixel labelling method segmented the textile images into different homogeneous colour regions sharing similar RGB colour features. It considered the spatial coherence in the neighbourhood system and thus provided more

smooth segmentation results than the standard clustering methods. Smoothness is an important element to obtain shape information for classification. Segmentation thus involves finding an optimal labelling which minimises the energy function of the MRF. To minimise the energy function, a global method, α - expansion, and a local method, ICM, were used. In both methods, parameter re-estimation can help to improve the accuracy. These algorithms were initialised based on a Gaussian mixture model (GMM). Compared with GMM, these methods can provide more smooth results but more slowly than GMM.

BIC was used to select the number of distinct class labels based on GMM. BIC based on GMM usually overestimated the number of class labels. However, this did not always result in $\alpha-$expansion with parameter re-estimation obtaining a less accurate result. Although the worst case is 37.53%, the best case is only 5.52% pixels mismatched to ground-truth labelling using $\alpha-$expansion (BIC).

The segmentation results were evaluated by a quantitative method which can deal with the case when a segmentation result has different number of labels from the ground truth. An error rate of mismatching segmentation labels using MRF pixel labelling to ground truth label was obtained. For all the images tested, $\alpha-$expansion with parameter re-estimation was the most accurate and the most computationally expensive method. $\alpha-$expansion without parameter re-estimation gave similar performance to ICM with parameter re-estimation. It shows that after each $\alpha-$expansion (new labelling obtained), model parameter estimation is important since the change of labelling results in the change of model parameter.

The results showed that MRF multi-pixel labelling model was an appropriate model for textile segmentation. The segmentation results were smooth and contained rich shape information, which is important to region-based classification or retrieval. The MRF segmentation results were used for image representation in Chapter 3.

5.1.2 Textile design representation

Chapter 3 overviewed the popular bag of words model, and summarised the advantages and disadvantages of this kind of model. Motivated by bag of words models, a novel image representation, bag of shapes, was proposed in Chapter 3. Visual words in this model were a collection of segmented regions. These regions were obtained by extracting connected components from MRF pixel labelling results. Each of the components was described using a 48-dimensional Generic Fourier Descriptor (GFD). There was no descriptor quantisation in this model. Images were represented by using a feature matrix in which each row was one shape descriptor.

One of the main drawbacks of the bag of words model is the lack of information about the spatial relationships between visual words. Region label graphs and their complements were constructed in order to capture the spatial information. This graph based method was based on MRF pixel labelling as well. Segmented regions with the same label were treated as a group and each group was associated uniquely with a vertex in an undirected, weighted graph. Each region group was represented as a bag of shape descriptors.

Edges in the graph denoted either the extent to which the groups' regions were spatially adjacent or the dissimilarity of their respective bags of shapes. It was tractable to create such graphs because the number of the vertices was not big. Once the graphs were constructed, their complements were obtained directly. Series of unweighted graphs were generated from the weighted graphs via edge removal. For the series of graphs, the corresponding sequences of chromatic number and domination number can be obtained. Those edge weights which made the chromatic number and domination number change were considered as graph features. The image feature was the linear combination of the shape features which was the histogram of codewords calculated from the bag of shapes

model, and the graph features extracted from two different graphs were different in edge meaning. The obtained compact feature vectors can be the input to standard machine learning methods, such as SVM.

5.1.3 Textile design classification

The image representations were investigated in the application of textile design classification. Images were from the Liberty Art Fabrics dataset and seven first level text descriptors were assigned to each image to compare the human labelling with the computer classification.

For the bag of shapes model, a simple yet effective classification scheme based on nearest neighbour matching with Naive Bayes assumption (NBNN) was used for classification. It requires no learning stage, or feature quantisation, and its classification performance ranks among the top learning based classification methods in the literature.

Based on NBNN, using bag of shapes model (MRF-GFD) obtained similar accuracy to bags of SIFT features. In the case of using MRF-GFD, not all of the shapes were used in the representation. The classification reached the peak, when the bound number of shapes was set to 150 empirically, namely the top 150 shapes sorted in an ascending order of their area were used for image representation. But it reduced gradually after 150. In the case of using bags of SIFT features, local regions were evenly or randomly sampled from images. With the number of local regions increased, the classification accuracy improved gradually but the execution time increased as well. Balancing accuracy and time, the number of local regions was set to 500. SIFT had more dimension than GFD, so this made bags of shapes model have the advantage of time efficiency. Actually, GFD was two orders of magnitude faster than SIFT.

To represent spatial relationships, region label graphs and complements were

constructed in different ways. One is using dissimilarity of groups' of regions as edges of graphs, and another one is using the arc length of common boundary between groups' of regions as edges of graphs. Different graph feature sets were obtained by calculating chromatic number and domination number. The graph features were tested in textile design classification, and compared in bag of shapes and bag of SIFT features using a linear SVM classifier. Generally, in both cases, the use of graph features can improve the accuracy of classification. It is better than using GFD or SIFT features only. The accuracy of using graph features from calculating domination number performed slightly better than that of using chromatic number. Among all of the graph features, the combined use of graphs with edges representing adjacency and dissimilarity gave classification accuracy at least as good as other feature sets.

For both bags of shapes and bags of SIFT models, classification based on the NBNN framework can achieve similar performance compared to a linear SVM classifier, but with utilisation of graph features, linear SVM classifier performs better than NBNN in both cases, but not significantly.

5.2 Future directions

5.2.1 Textile design segmentation

In Chapter 2, the textile design segmentation problem was solved using MRF pixel labelling by finding an optimal labelling to minimise the MRF energy function. To minimise the MRF energy function, α - expansion, a graph cuts algorithm was used and can obtain smooth segmentation results. Compared with GMM and ICM, it is slow to use this algorithm. It takes about $15s$ to process images with size of 500×500, about ten times slower than GMM and ICM.

Komodakis *et al.*[67] proposed a new method which is based on linear program-

ming theory to address minimisation of MRF energy functions. This method generalises prior state-of-the-art methods, such as α - expansion, and can offer a substantial speedup of one order of magnitude over existing methods. Therefore, it would be worth trying this method for efficiency in the future. The code is available at [66].

One issue in segmentation is how to decide the number of colour labels. The Liberty Art Fabrics database contains information of how many colour dyes are used in the design. It is also interesting to try to automatically get the number of colours. In our experiments, the number was decided using BIC based on a Gaussian mixture model, but the number was always overestimated. It would be worth trying other methods for model order selection and using methods which estimate the number based on the Markov Random Field model. These methods are usually very expensive, e.g. Reversible Jump MCMC sampling [61].

5.2.2 Textile design representation

In Chapter 3, images were represented by using bag of shapes models which were based on MRF pixel labelling results. Due to efficiency, only the largest segmented regions' features were used and each region feature was supposed to have the same weight.

To capture the spatial information, undirected weighted graphs were constructed and graph features were extracted by calculating chromatic number and domination number from graphs. The features were fixed dimensional vectors combining shape features and graph features, and can be used as the input to any standard machine learning classification method. An alternative to deriving feature vectors from a graph is to employ graph kernel methods [60, 51]. Given that the graphs constructed in Section 3.4 have relatively few vertices, kernels could be designed without so much regard to computational complexity as is needed when dealing with large graphs. It would be interesting to compare this approach in

future work.

Another promising direction is to continue to explore much more efficient features from region label graphs, since using chromatic number and domination number is just an intuitively simple attempt, there are certainly many other ways of using the region label graphs. Constructing region label graphs utilises the colour cues. (Images are segmented into different homogeneous regions according to similar colours). This image representation can be extended to the use of general images by using other cues. For example, a general segmentation problem is to find objects in images, and MRF model can also used for object-oriented segmentation tasks. Different objects could be the nodes of the graphs and edges could be the similarity or the common boundaries of the objects.

5.2.3 Textile design classification

To test the effectiveness of the image representations, these image representations were applied in classification task. First-level text descriptors were assigned to images based on design type and visual content. Each image was assigned one single text descriptor, so images were classified according to these text descriptors. This classification labelling was used as ground-truth to compare with computer classification results. Seven text descriptors were generally termed "style" within the archive database: floral, paisley, stripe, leaf, geometric, spot and check. However, one of the text descriptor, geometric, defined a broad design style that included images with more variable content but with geometric attributes (e.g., angles, lines, simple shapes). Some terms (floral, paisley, and leaf) referred to the design content of the image in familiar textile design terms (especially in the cases of floral and paisley); others (stripe, spot, and check) provided both a description of style and visual content. All of the elements may result in confusion in textile design classification and the varied ways of describing the images should be refined.

Taking into account these reasons, a system has been implemented for image annotation as shown in Figure 5.1. In the system, users can label images with more than one text descriptor according to visual content contained in images. The experimental data used in Chapter 4 were relabelled by Dr. Ward manually according to the visual content elements in images. Nine text descriptors were: flower, stem, leaf, paisley, plaid, check, dot, stripe, and circle-like and each text descriptor has two types: abstract or non-abstract. Besides, the importance order of the text descriptors was required to tell the importance of the content contained in images. For the image shown in Figure 5.1, Dr. Ward marked the text descriptors in an order of leaf, stem, flower and circle-like. The potential use of importance order of text descriptors may be helpful for automatic tagging and annotation. In the future experiments, each testing image will be assigned more than one or at least one class label automatically, and the results will be evaluated by comparing with the labels assigned by Dr. Ward's labelling manually through the annotation system.

Wang *et al.* [129], also in FABRIC project, have done research on image visualisation which is essential for browsing and exploring the image content. They use high-entropy layout distribution (HELD) to arrange collections of images for display with dependencies both on content of the images and the layout shape [129]. The image content they consider are colour and texture, and features used are 512-dimensional colour histograms and Gabor texture features. One direct experiment that can be run on their system is to input the linear combination of shape and graph features proposed in this thesis and compare the layout result with that using colour and texture features. The proposed features can also be applied on the interactive image retrieval system [49] which was also developed in the FABRIC project. User's feedback obtained via interaction with 2D image layouts provides qualitative constraints that can be used to adapt the distance metric for image retrieval. In this system, three type of low-level features were used: a 36-dimensional colour histogram, an 18-dimensional texture feature ba-

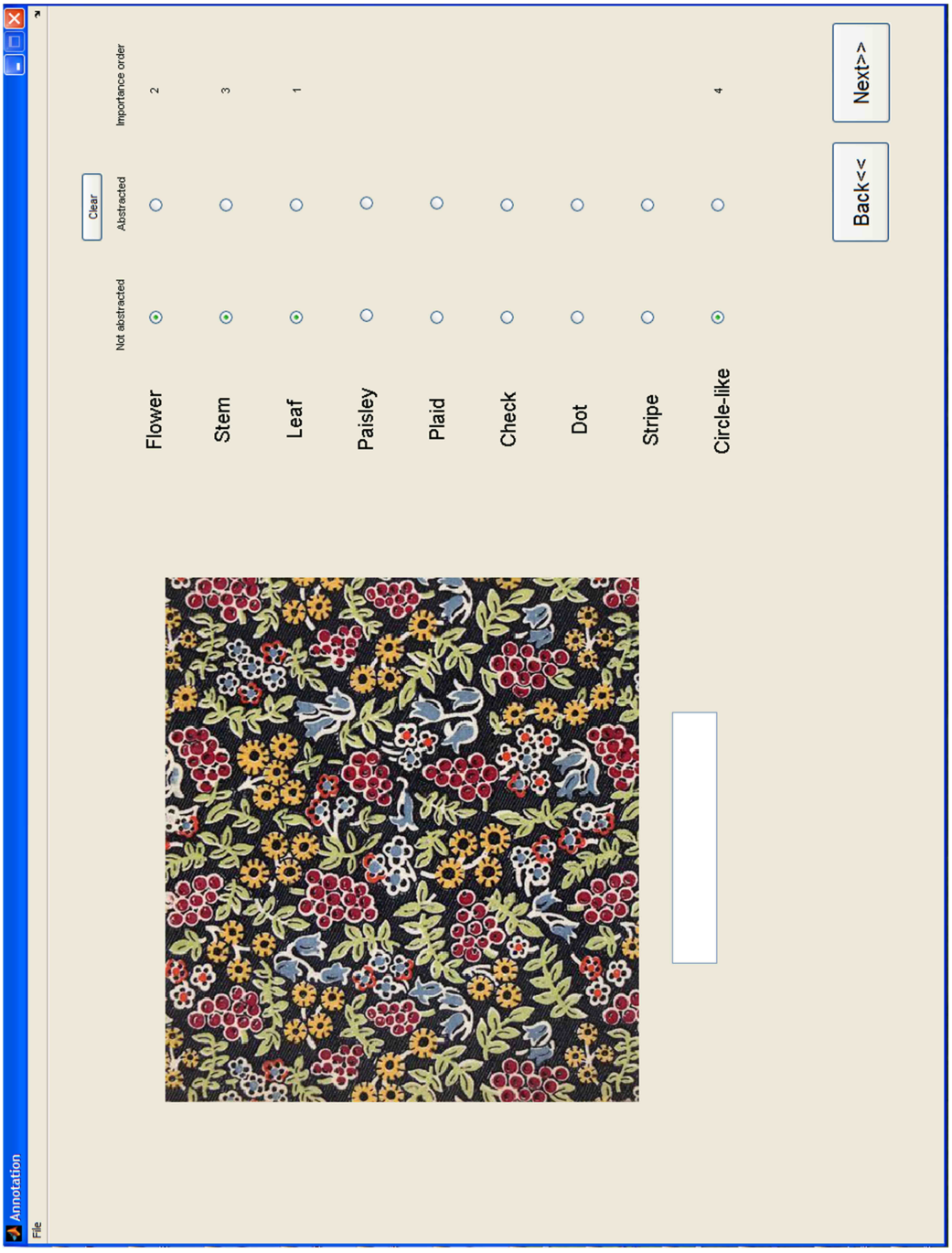


Figure 5.1: Annotation system for assigning each image multiple labels. (Image courtesy of Liberty Art Fabrics. ©Copyright protected; Reproduction not permitted.)

sed on a wavelet transformation [113], and a 144-dimensional colour correlogram. It would be worth using the proposed features instead to compare the retrieval results.

The following page is left intentionally blank.

Bibliography

[1] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space. *Database Theory - ICDT 2001, Proceedings*, 1973:420–434, 2001.

[2] H. Akaike. A new look at the statistical identification model. *IEEE Transactions on Auto. Control*, 19:716–723, 1974.

[3] S. Aksoy, K. Koperski, C. Tusk, G. Marchisio, and J. C. Tilton. Learning Bayesian classifiers for scene classification with a visual grammar. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):581–589, 2005.

[4] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. *IEEE Conference on Computer Vision and Pattern Recognition, Vols 1-4*, pages 2294–2301, 2009.

[5] V. Athitsos, J. Alon, and S. Sclaroff. Efficient nearest neighbor classification using a cascade of approximate similarity measures. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:486–493, 2005.

[6] S. T. Barnard. Stochastic stereo matching over scale. *International Journal of Computer Vision*, 3(1):17–32, 1989.

[7] S. Battiato, G. M. Farinella, G. Gallo, and D. Ravi. Scene categorization using bag of textons on spatial hierarchy. *IEEE International Conference on Image Processing*, 1-5:2536–2539, 2008.

[8] J Besag. On the statistical analysis of dirty picutures. *Journal of Royal Statistical Sociely, Series B*, 48(3):259–302, 1986.

[9] J. B. Bi, Y. X. Chen, and J. Z. Wang. A sparse support vector machine approach to region-based image categorization. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:1121–1128, 2005.

[10] C. M. Bishop. *Pattern Recognition and Machine Learning*. New York: Springer-Verlag, 2006.

[11] A. Blake and A. Zisserman. *Visual Reconstruction*. Cambridge, MA: MIT Press, 1987.

[12] I. Bloch. Fuzzy spatial relationships for image processing and interpretation: A review. *Image and Vision Computing*, 23(2):89–110, 2005.

[13] M. Bober. MPEG-7 visual shape descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:716–719, 2001.

[14] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. *IEEE Conference on Computer Vision and Pattern Recognition*, 1-2:1992–1999, 2008.

[15] M. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[16] M. R. Boutell, J. Luo, and C. M. Brown. Factor graphs for region-based whole-scene classification. *IEEE Conference on Computer Vision and Pattern Recognition, Semantic Leaning Workshop*, 2006.

[17] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

[18] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9): 1124–1137, 2004.

[19] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–655, 1998.

[20] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[21] C. Bron and J. Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Communications of the ACM*, 16 (9):575–577, 1973.

[22] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[23] C-C Chang and C-J Lin. LIBSVM-a library for support vector machines. http://www.csie.ntu.edu.tw/ cjlin/libsvm/, September 2010.

[24] Y. X. Chen and J. Z. Wang. Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 5:913–939, 2004.

[25] A. Choksuriwong, B. Emile, C. Rosenberger, and H. Laurent. Comparative study of global invariant descriptors for object recognition. *Journal of Electronic Imaging*, 17(2):023015, 2008.

[26] D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 750–755, 1997.

[27] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[28] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[29] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. *ECCV Workshop on Statistical learning in Computer Vision*, 2004.

[30] C. de Mauro, M. Diligenti, M. Gori, and M. Maggini. Similarity learning for graph-based image representations. *Pattern Recognition Letters*, 24(8): 1115–1122, 2003.

[31] A. Delong, A. Osokin, N. H. Isack, and Y. Boykov. Fast approximate energy minimization with label cost. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2173–2180, 2010.

[32] J. Demel. *Graphs and their Applications*. 1st Pragure: Academia, 2002.

[33] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via EM algorithm. *Journal of The Royal Statistical Society Series B-Methodological*, 39(1):1–38, 1977.

[34] Y. N. Deng and B. S. Manjunath. JESG-Segmentation of Color-Texture Regions in Images and Video. http://vision.ece.ucsb.edu/segmentation/jseg/, 1999.

[35] Y. N. Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):800–810, 2001.

[36] E. A. Dinic. Algorithm for solution of a problem of maximum flow in networks with power estimation. *Soviet Math. Dokl.*, 11:1277–1280, 1970.

[37] R. O. Duda, P. E. Hart, and D. G. S. *Pattern Classification*. John Wiley & Sons Inc, 2001.

[38] A. S El-Ghazal. *Multi-Technique Fusion for shape-Based Image Retrieval*. PhD thesis, University of Waterloo, Canada, 2009.

[39] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:524–531, 2005.

[40] J. Flusser. On the independence of rotation moment invariants. *Pattern Recognition*, 33:1405–1410, 2000.

[41] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[42] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[43] D. Gokalp and S. Aksoy. Scene classification using bag-of-regions representations. *IEEE Transcations on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[44] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *Journal of The ACM*, 35(4):921–940, 1988.

[45] P. J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[46] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a-posteriori estimation for binary images. *Journal Of The Royal Statistical Society Series B-Methodological*, 51(2):271–279, 1989.

[47] C. H. Gu, J. J. Lim, P. Arbelaez, and J. Malik. Recognition using regions. *IEEE Conference On Computer Vision And Pattern Recognition, Vols 1-4*, pages 1030–1037, 2009.

[48] J. M. Hammersley and P. Clifford. Markov field on finite graphs and lattices. *unipublished*, 1971.

[49] J. Han, S. J. McKenna, and R. Wang. Learning query-dependent distance metrics for interactive image retrieval. *Proceedings of 7th International Conference on Computer Vision Systems (ICVS)*, pages 374–383, 2009.

[50] F. Harary. *Graph Theory*. Reading, MA: Addison-Wesley, 1994.

[51] Z. Harchaoui and F. Bach. Image classification with segmentation graph kernels. *IEEE Conference On Computer Vision and Pattern Recognition*, 1-8:612–619, 2007.

[52] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Domination in Graphs-Advanced Topics*. New York: Dekker, 1998.

[53] C-W Hsu and C-J Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Netwrods*, 13(2):415–425, 2002.

[54] M. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT-8:179–182, 1962.

[55] H. Ishikawa and D. Geiger. Segmentation by grouping junctions. *IEEE Computer Society Conference On Computer Vision And Pattern Recognition, Proceedings*, pages 125–131, 1998.

[56] A. K. Jain and F. Farroknia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[57] A. K. Jain, R. P. W. Duin, and J. C. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[58] F. Jing, M. J. Li, H. J. Zhang, and B. Zhang. An efficient and effective region-based image retrieval framework. *IEEE Transactions on Image Processing*, 13(5):699–709, 2004.

[59] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. *IEEE International Conference on Computer Vision*, 1 and 2:604–610, 2005.

[60] H. Kashima, K Tsuda, and A. Inokuchi. *Kernels for Graphs.* MIT Press, 2004.

[61] Z. Kato. Segmentation of color images via reversible jump mcmc sampling. *Image and Vision Computing*, 26(3):361–371, 2008.

[62] Z. Kato and T. C. Pong. A markov random field image segmentation model for color texture images. *Image and Vison Computing*, 24: 1103–1114, 2006.

[63] W-Y. Kim and Y-S Kim. A region-based shape descriptor using zernike moments. *Signal Processing: Image Communication*, 16:95–102, 2000.

[64] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, 2006.

[65] V. Kolmogorov and R. Zabih. What energy functions can be minmized via graph cuts. *IEEE Transcations on Pattern Analysis and Machine Intelligence*, 26:147–159, 2004.

[66] N. Komodakis. FastPD MRF optimization code. http://www.csd.uoc.gr/ komod/FastPD/index.html.

[67] N. Komodakis, G. Tziritas, and N. Paragios. Performance vs computational efficiency for optimizing single and dynamic MRFs: Setting the state of the art with primal-dual strategies. *Computer Vision and Image Understanding*, 112(1):14–29, 2008.

[68] A. Kumar and C. Sminchisescu. Support kernel machines for object recognition. *International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

[69] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Solving Markov random fields using second order cone programing. *IEEE Conference on Computer Vision and Pattern Recognition*, I:1045–1052, 2006.

[70] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. *IEEE International Conference on Computer Vision*, 2:1284–1291, 2005.

[71] I. Kunttu and L. Lepisto. Shape-based retrieval of industrial surface defects using angular radius fourier descriptor. *LET Image Processing*, 1: 231–236, 2007.

[72] M. Kutil, P. Sucha, M. Sojka, and Z. Hanzalek. Torsche scheduling toolbox for matlab. http://rtime.felk.cvut.cz/scheduling-toolbox/.

[73] D. A. Langan, J. W. Modestino, and J. Zhang. Cluster validation for unsupervised stochastic model-based image segmentation. *IEEE Transaction on Image Processing*, 7(2):180–195, 1998.

[74] D. Larlus and F. Jurie. Latent mixture vocabularies for object categorization and segmentation. *Image and Vision Computing*, 27(5): 523–534, 2009.

[75] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.

[76] T Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43 (1):29–44, 2001.

[77] J. Li, J. Z. Wang, and G. Wiederhold. Irm: integrated region matching for image retrieval. *Proceedings of the ACM International Conference on Multimedia*, pages 147–156, 2000.

[78] S. Z. Li. *Markov Random Field Modeling in Computer Vision.* Springer-Verlag, 1995.

[79] Y. P. Li, M. J. Kyan, and L. Guan. Improving shape-based CBIR for natural image content using a modified GFD. *Image Analysis and Recognition*, 3656:593–600, 2005.

[80] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[81] M. Maire, P. Arbelaez, C. Fowlkes, and M. Malik. Using contours to detect and localize junctions in natural image. *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[82] P Manandhar. Polar to/from rectangular transform of images. http://www.mathworks.com/matlabcentral/fileexchange/17933-polar-tofrom-rectangular-transform-of-images, 2007.

[83] J. L. Marroquin, F. A. Velasco, and M. Nakamura. Gauss-Markov measure field models for low-level vision. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 23:337–348, 2001.

[84] J. L. Marroquin, E. Arce, and S. Botello. Hidden Markov measure field models for image segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 25:1380–1387, 2003.

[85] J. M. Martinez. MPEG-7 overview. http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm, October 2004.

[86] A. McCallum and K Nigam. A comparison of event models for naive Bayes text classification. *AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48, 1998.

[87] F. Meyer. Hierarchies of partitions and morphological segmentation. *Scale-Space and Morphology In Computer Vision*, 2106:161–182, 2001.

[88] B. Micusik and T. Pajdla. Multi-label image segmentation via max-sum solver. *IEEE Conference on Computer vision and Pattern Recognition*, 2007.

[89] K. Mikolajczyk. Affine covariant features. http://www.robots.ox.ac.uk/ vgg/research/affine/descriptors.html, July 2007.

[90] K. Mikolajczyk and C Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[91] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005.

[92] D. M. Mount and S. Arya. Ann: A library for approximate nearest neighbor searching. http://www.cs.umd.edu/~mount/ANN/, January 2010.

[93] R. Mukundam. Radial Tchebichef invariants for pattern recognition. *IEEE Tencon Conference*, pages 2098–2103, 2005.

[94] R. Mukundam, S. H. Ong, and P. A. Lee. Image analysis by Tchebichef moments. *IEEE Transactions on Image Processing*, 10:1357–1364, 2001.

[95] N. V. Nguyen, A. Boucher, J-M. Ogier, and S. Tabbone. Region-based semi-automatic annotation using the bag of words representation of the keywords. *International Conference on Image and Graphics*, 2009.

[96] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. *IJCAI Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

[97] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. *IEEE Conference on Computer Vision and Pattern Recognition*, 27(10): 1615–1630, 2005.

[98] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. *European Conference on Computer Vision*, pages 490–503, 2006.

[99] H. Permuter, J. Francos, and I. Jermyn. A study of Gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recognition*, 39(4):695–706, 2006.

[100] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. *IEEE Conference of Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[101] R. Potts. Some generalized order-disorder transformation. *Proc. Cambridge Philosophical Soc.*, 48:106–109, 1952.

[102] G. Qiu, X. Feng, and J. Feng. Compressing histogram representations for automatic colour photo categorization. *Pattern Recognition*, 37(11): 2177–2193, 2004.

[103] P. Quelhas, F. Monay, J. M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool. Modeling scenes with local descriptors and latent aspects. *IEEE International Conference on Computer Vision*, 1 and 2:883–890, 2005.

[104] M. Rivera, O. Ocegueda, and J. L. Marroquin. Entropy-controlled quadratic Markov measure field models for efficient image segmentation. *IEEE Transcations on Image Processing*, 16:3047–3057, 2007.

[105] S. J. Roberts, D. Husmeier, I. Rezek, and W. Penny. Bayesian approaches to Gaussian mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1133–1142, November 1998.

[106] S. Roy and I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. *International Conference on Computer Vision*, pages 492–499, 1998.

[107] S. Roy and V. Govindu. MRF solutions for probabilistic optical flow formulations. *International Conference on Pattern Recognition*, 3: 1041–1047, 2000.

[108] E. R. Scheinerman. Matgraph: A Matlab toolbox for graph theory. http://www.ams.jhu.edu/ ers/matgraph/, 2005.

[109] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

[110] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. *IEEE International Conference on Computer Vision*, I and II:1470–1477, 2003.

[111] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. *Tenth IEEE International Conference On Computer Vision*, 1 and 2:370–377, 2005.

[112] S Skiena. *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica.* Reading, MA: Addison-Wesley, 1990.

[113] J. R. Smith and S. F. Chang. Automated binary texture feature sets for image retrieval. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2239–2242, 1996.

[114] M. Stricker and A. Dimai. Similarity of color image. *Storage and Retrieval for Image and Video Database III (W. Niblack and R. C. Jain, Eds), Proc. SPIE*, 2420:381–392, 1995.

[115] M. J. Swain and B. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[116] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. MRF minimization. http://vision.middlebury.edu/MRF/, 2007.

[117] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):1068–1080, 2008.

[118] M Teague. Image analysis via the general theory of moments. *Journal of the Optical Society of America*, 70:920–930, 1980.

[119] C. H. Teh and R. T. Chin. On image analysis by the methods of moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):496–513, 1988.

[120] P. Tirilly, V. Claveau, and P. Gros. Language modeling for bag-of-visual words image categorization. *International Conference on Content-based Image and Video Retrieval*, pages 249–258, 2008.

[121] D. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions*. Hoboken, NJ: Wiley, 1985.

[122] C-F Tsai and W-C Lin. A comparative study of global and local feature representation in image database categorization. *International Joint Conference on INC, IMS and IDC*, pages 1563–1566, 2009.

[123] T. Tuytelaars and C. Schmid. Vector quantizing feature space with a regular lattice. *IEEE International Conference on Computer Vision*, 1-6: 754–761, 2007.

[124] J. M. Valiente, M. C. Carretero, J. M. Gómis, and F. Albert. Image processing tool for the purpose of textile fabric modeling. In *International Conference on Designs Tools and Methods*, 2001.

[125] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *IEEE International Conference On Computer Vision*, 2007.

[126] O. Veksler. Image segmentation by nested cuts. *IEEE Conference on Computer Vision and Pattern Recognition, Proceedings*, I:339–344, 2000.

[127] O. Veksler and A. Delong. http://vision.csd.uwo.ca/code/.

[128] D. Molder C. Vizitiu, C-I. Munteanu. A new version of flusser moment set for pattern feature extraction. *WSEAS Transactions on Information Science and Applications*, 5:920–930, 2008.

[129] R. Wang, S. J. McKenna, J. Han, and A. A. Ward. Visualizing image collections using high-entropy layout distributions. *IEEE Transcations on Multimedia*, 12(8):803–813, 2010.

[130] T. Werner. A linear programming approach to max-sum problem: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29, No. 7:1165–1179, 2007.

[131] J. Willamowski, D. Arregui, G. Csurka, and C. R. Dance. Categorizaing nice visual classes using local appearance descriptors. *ICPR Workshop on Learning for Adaptable Visual System*, 2004.

[132] C. S. Won and H. Derin. Unsupervised segmentation of noisy and textured images using markov random fields. *Computer Graphics and Image Processing: Graph Models and Image Processing*, 52(4):208–328, 1992.

[133] Y. Wu, X Yang, and K. L. Chan. Unsupervised colour image segmentation based on gaussian mixture model. *Proceedings of the Joint Conference of the 4th International Conference on Information, Communications and Signal Processing*, 1:541–544, 2003.

[134] H. Xu, D. Li. Geometric moment invariants. *Pattern Recognition*, 41: 240–249, 2008.

[135] R. Xu and D. Wunsch II. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16, NO. 3:645–678, 2005.

[136] J. Yang, Y. G. Jiang, A. G. Hauptmann, and C. W. Ngo. Evaluating bag-of-visual-words representations in scene classification. *Proceedings of the international workshop on Multimedia Information Retrieval*, pages 197–206, 2007.

[137] P. T. P. Yap and R. S. H. Ong. Image analysis by Krawtchouk moment. *IEEE Transactions on Image Processing*, 12:1367–1377, 2003.

[138] J. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. *Advances in Neural Information Processing Systems*, pages 689–695, 2000.

[139] J. Yuan, Y. Wu, and M. Yang. Discovery of collocation patterns: from visual words to visual phrases. *IEEE Conference on Computer Vision and Pattern Recognition*, 1-8:1930–1937, 2007.

[140] R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. *IEEE Conference on Computer Vision and Pattern Recognition*, 2: 473–444, 2004.

[141] D. Zhang and G Lu. Evaluation of MPEG-7 shape descriptors against other shape descriptors. *Multimedia System*, 9:15–30, 2003.

[142] D. S Zhang. *Image Retrieval Based on Shape*. PhD thesis, Monash University, 2002.

[143] D. S. Zhang and G. J. Lu. Shape-based image retrieval using generic Fourier descriptor. *Signal Processing-Image Communication*, 17(10): 825–848, 2002.

[144] H. Zhang, A. C. Berg, M. Maire, and J. Malik. Discriminative nearest neighbor classification for visual category recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:2126–2136, 2006.

[145] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007.

[146] Q. F. Zheng and W. Gao. Constructing visual phrases for effective and efficient object-based image retrieval. *ACM Transactions on Multimedia Computing Communications and Applications*, 5(1), 2008.

[147] H. Zhu, H. Shu, J. Liang, L. Luo, and J. L. Coatrieux. Image analysis by discrete orthogonal Racah moments. *Signal Processing*, 87:687–708, 2007.

[148] H. Zhu, J. Shu, H. Zhou, L. Luo, and J. L. Coatrieux. Image analysis by discrete othogonal dual Hahn moments. *Pattern Recognition Letters*, 28: 1688–1704, 2007.